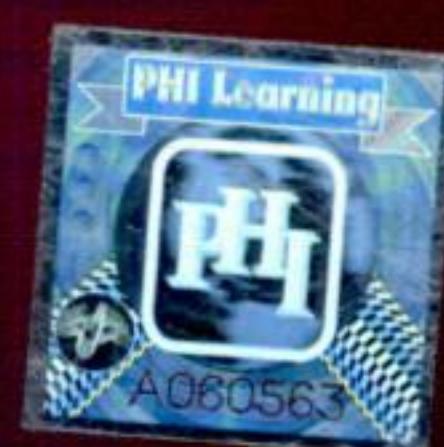


Cryptography and Information Security



V.K. Pachghare



Rs. 275.00

CRYPTOGRAPHY AND INFORMATION SECURITY
V.K. Pachghare

© 2009 by PHI Learning Private Limited, New Delhi. All rights reserved. No part of this book may be reproduced in any form, by mimeograph or any other means, without permission in writing from the publisher.

ISBN-978-81-203-3521-9

The export rights of this book are vested solely with the publisher.

Published by Asoke K. Ghosh, PHI Learning Private Limited, M-97, Connaught Circus, New Delhi-110001 and Printed by Mudrak, 30-A, Patparganj, Delhi-110091.

Contents

Preface

xv

1. Introduction 1-10

1.1 Security	1
1.2 Elements of Information Security	2
1.2.1 Confidentiality	2
1.2.2 Integrity	2
1.2.3 Availability	3
1.3 Security Policy	3
1.4 Security Techniques	4
1.5 Steps for Better Security	5
1.6 Category of Computer Security	6
1.7 The Operational Model of Network Security	7
1.8 Basic Network Security Terminology	8
Summary	10
Exercises	10

2. Data Encryption Techniques 11-31

2.1 Introduction	11
2.2 Encryption Methods	12
2.2.1 Symmetric Encryption	12
2.3 Cryptography	13
2.4 Cryptanalysis	14
2.5 Substitution Ciphers	15
2.5.1 The Caesar Cipher	15
2.5.2 Monoalphabetic Ciphers	16
2.5.3 Playfair Cipher	17
2.5.4 The Hill Cipher	19
2.5.5 Polyalphabetic Ciphers	22
2.5.6 One-time Pad	24

2.6	Transposition Ciphers	26
2.6.1	Single Columnar Transposition	26
2.6.2	Double Columnar Transposition	27
2.7	Steganography	28
2.7.1	Uses of Steganography	28
2.7.2	Steganography and Security	29
	Summary	30
	Exercises	30
3.	Data Encryption Standards	32-73
3.1	Block Ciphers	32
3.2	Block Cipher Modes of Operation	32
3.2.1	Electronic Code Book Mode	33
3.2.2	Cipher Block Chaining Mode	34
3.2.3	Feedback Modes	35
3.2.4	Counter Mode	38
3.3	Feistel Ciphers	40
3.4	Data Encryption Standard	42
3.4.1	Works of DES	43
3.4.2	Cracking DES	67
3.5	Triple DES	67
3.5.1	Working of Triple DES	68
3.5.2	Modes of Operation	69
3.6	DES Design Criteria	69
3.7	Side Channel Attacks	71
3.8	Other Block Ciphers	71
3.9	Differential Cryptanalysis	72
3.10	Linear Cryptanalysis	72
	Summary	72
	Exercises	73
4.	Advanced Encryption Standard	74-86
4.1	Introduction	74
4.2	Advanced Encryption Standard	75
4.3	Overview of Rijndael	75
4.4	Optimization of the Cipher	83
4.5	Advantages and Limitations of Rijndael	84
4.6	Comparison of AES with Other Ciphers	84
	Summary	85
	Exercises	86
5.	Symmetric Ciphers	87-105
5.1	Blowfish Encryption Algorithm	87
5.1.1	Generating the Subkeys	87
5.1.2	The Algorithm	88
5.1.3	The Feistel Structure of Blowfish	90
5.1.4	Cryptanalysis of Blowfish	91
5.1.5	Blowfish in Practice	91

5.2	RC5	92
5.2.1	Characteristics of RC5	94
5.2.2	Parameters	94
5.2.3	Cipher Modes in RC5	94
5.3	RC4	95
5.3.1	Design Considerations	96
5.3.2	Characteristics	96
5.3.3	Algorithms	96
5.3.4	RC4 Security and Efficiency	97
5.3.5	Weaknesses of RC4	98
5.3.6	The Use of RC4	98
5.4	RC6	98
5.4.1	Description of RC6	99
5.4.2	Basic Operations	100
5.4.3	Block Diagram	100
5.5	Comparison between RC6 and RC5	100
5.6	IDEA	101
5.6.1	Working of IDEA	101
5.6.2	Decryption	103
5.6.3	Security	103
5.6.4	Weaknesses of IDEA	103
Summary		104
Exercises		105

6. Number Theory 106–117

6.1	Prime Number	106
6.2	Fermat's Theorem	107
6.3	Euler's Theorem	107
6.4	Primality Test	108
6.4.1	Naïve Methods	109
6.4.2	Probabilistic Tests	109
6.4.3	Fermat Primality Test	110
6.4.4	Miller–Rabin Primality Test	110
6.4.5	Leonard Adleman and Huang Primality Test	110
6.4.6	Fast Deterministic Tests	110
6.4.7	Number-Theoretic Methods	111
6.5	Chinese Remainder Theorem	111
6.6	Discrete Logarithms	114
6.6.1	Index Calculus Algorithm	114
6.6.2	Baby-step Giant-step	116
Summary		116
Exercises		117

7. Public Key Cryptosystems 118–137

7.1	Introduction	118
7.2	Public Key Encryption	119
7.2.1	Authentication	121
7.2.2	Key Length and Encryption Strength	124

7.2.3	Applications of Public Key Cryptography	124
7.2.4	Strength and Weakness of Public Key	125
7.2.5	Comparison of Asymmetric Encryption and Symmetric Encryption	125
7.3	The RSA Algorithm	125
7.3.1	Key Generation Algorithm	126
7.3.2	Key Length	128
7.3.3	Security	129
7.4	Timing Attacks	130
7.4.1	Possible Defences against Timing Attack	133
Problems		134
Summary		134
Exercises		137
8.	Key Management.....	138-161
8.1	Key Distribution	138
8.2	Diffie—Hellman Key Exchange	141
8.2.1	Description	142
8.2.2	Security	144
8.2.3	Man-in-the-Middle Attack	145
8.2.4	Authentication	145
8.3	Elliptic Curve Arithmetic	145
8.3.1	Elliptic Curve Groups over Real Numbers	146
8.3.2	Elliptic Curve Addition: A Geometric Approach	146
8.3.3	Elliptic Curve Addition: An Algebraic Approach	149
8.3.4	Elliptic Curve Groups over F_p	149
8.3.5	Arithmetic in an Elliptic Curve Group over F_p	151
8.3.6	Elliptic Curve Groups over F_{2^m}	151
8.3.7	Arithmetic in an Elliptic Curve Group over F_{2^m}	153
8.4	Elliptic Curve Cryptography	154
8.4.1	Elliptic Curve Diffie—Hellman	154
8.4.2	Key Establishment Protocol	154
8.5	Elliptic Curve Security and Efficiency	155
8.6	Zero-Knowledge Proof	157
Problems		160
Summary		161
Exercises		161
9.	Authentication	162-203
9.1	Introduction	162
9.1.1	Objectives	163
9.1.2	Measurements	163
9.2	Authentication Methods	164
9.2.1	Password-Based Authentication Method	164
9.2.2	Two-factor Authentication Method	167
9.2.3	Biometric Authentication Method	167
9.2.4	Extensible Authentication Protocol	170

9.3	Message Digest	172
9.3.1	MD2	172
9.3.2	MD4	173
9.3.3	MD5	173
9.3.4	SHA-1	176
9.3.5	HMAC	182
9.3.6	RIPEMD-160	183
9.4	Kerberos	184
9.4.1	The Basics of Kerberos	185
9.4.2	Kerberos Ticket-granting Approach	188
9.4.3	The Ticket Granting Server	189
9.4.4	Kerberos Third-party Authentication Model	190
9.4.5	Kerberos Authentication Model: Definitions and Notational Conventions	191
9.4.6	Kerberos Authentication Model	192
9.4.7	Cross-Realm Authentication	194
9.4.8	Kerberos and Public Key Cryptography	195
9.4.9	Advantages of Kerberos	196
9.4.10	Weaknesses of Kerberos	197
9.4.11	Attacks on Kerberos	198
9.4.12	Applications and Limitations of Kerberos	199
9.4.13	Competition to Kerberos	199
9.5	X.509 Authentication Service	200
	Summary	202
	Exercises	202

10. Digital Signatures.....204-215

10.1	Introduction	204
10.1.1	Implementation of Digital Signatures	206
10.1.2	Association of Digital Signatures and Encryption	206
10.1.3	Using Different Key Pairs for Signing and Encryption	207
10.2	Digital Signature Algorithms	208
10.2.1	Digital Signature Algorithm	209
10.2.2	ElGamal Signature	211
10.2.3	Elliptic Curve DSA	212
10.2.4	Signature Generation Algorithm	212
10.2.5	Signature Verification Algorithm	212
10.3	Digital Signature Standard (DSS)	213
10.3.1	Applications	214
10.4	Authentication Protocols	214
	Summary	215
	Exercises	215

11. Electronic Mail Security.....216-238

11.1	Pretty Good Privacy (PGP)	216
11.1.1	Need of PGP	217
11.1.2	Working of PGP	219

11.1.3 PGP Encryption Applications	220
11.1.4 PGP: Backdoors and Key Escrow	221
11.1.5 PGP Security Quality	223
11.2 S/MIME	223
11.3 MIME	224
11.3.1 MIME Headers	224
11.3.2 MIME Transfer-Encoding Header Field	230
11.4 History of S/MIME	232
11.4.1 Working of S/MIME	234
11.4.2 Applications of S/MIME	236
11.5 Comparison of PGP and S/MIME	237
Summary	237
Exercises	237

12. IP Security 239–266

12.1 The IP Security Architecture	239
12.1.1 Strengths of IPsec	239
12.1.2 Applications of IPSec	240
12.1.3 Benefits of IPSec	241
12.1.4 Overview of IP Security	241
12.1.5 Working of IPsec	242
10.2 IPsec, IPv4, and IPv6	242
12.2.1 IPsec Protocols and Operations	243
12.3 The Authentication Header (AH) Protocol	244
12.3.1 Using AH Header	245
12.3.2 Transport Mode	245
12.3.3 Tunnel Mode	246
12.4 The Encapsulating Security Payload (ESP) Protocol	247
12.4.1 Using ESP Header	249
12.4.2 Using IPsec Tunnels	250
12.4.3 Cryptographic Algorithms	251
12.4.4 Implementing and Deploying IPsec	251
12.4.5 Usage	251
12.5 The ISAKMP Protocol	253
12.5.1 Overview	253
12.5.2 Terms and Definitions	253
12.5.3 Security Association Negotiation	254
12.5.4 ISAKMP Payloads	255
12.5.5 ISAKMP Exchange Types	257
12.5.6 Security Association Establishment	260
12.6 The OAKLEY Key Determination Protocol	263
12.6.1 Overview	263
12.7 The Key Exchange Protocol	264
Summary	265
Exercises	266

13. Web Security 267–280

- 13.1 Secure Socket Layer 267
- 13.2 SSL Session and Connection 269
- 13.3 The SSL Record Protocol 270
- 13.4 The SSL in Practice 275
- 13.5 Secure Electronic Transactions 277
 - 13.5.1 Importance of SET 277
 - 13.5.2 SET Mechanism 278
 - 13.5.3 Establishment of the Framework 278
 - 13.5.4 Conduct of a Payment Transaction 278
 - 13.5.5 Strengths of SET 279
 - 13.5.6 Weaknesses of SET 279
- Summary 279
- Exercises 280

14. Intrusion 281–305

- 14.1 Introduction 281
- 14.2 Intrusion Detection 282
- 14.3 Intrusion Detection System 282
 - 14.3.1 The Need for Intrusion Detection Systems 283
 - 14.3.2 Classification of Intrusion Detection Systems 285
- 14.4 Anomaly Detection Systems 287
 - 14.4.1 Statistical Approaches 287
 - 14.4.2 Predictive Pattern Generation 288
- 14.5 Misuse Detection Systems 288
- 14.6 Rule-Based Intrusion Detection 292
- 14.7 Distributed Intrusion Detection 293**
 - 14.7.1 Overview 293**
 - 14.7.2 Advantages of a dIDS 295**
 - 14.7.3 Incident Analysis with dIDS 295**
 - 14.7.4 Analysis Using Aggregation 295**
- 14.8 Base-Rate Fallacy 296**
 - 14.8.1 Basic Frequency Assumptions 297**
 - 14.8.2 Honeypots 297**
- 14.9 Password Management Best Practices 298
- 14.10 Limitations of Intrusion Detection Systems 304
- Summary 305
- Exercises 305

15. Malicious Software 306–328

- 15.1 Malicious Code 306**
- 15.2 Viruses 306**
 - 15.2.1 Types of Viruses 307**
 - 15.2.2 Working of Anti-Virus Software 309
 - 15.2.3 Methods to Avoid Detection 310**

15.3 Worms	313
15.3.1 Historical Background	314
15.3.2 Different Types of Computer Worms	314
15.3.3 Protecting against Computer Worms	316
15.4 Trojans	316
15.4.1 Trojan Horses	316
15.5 Spyware	317
15.6 Best Practices	317
15.7 Digital Immune System	318
15.7.1 Behaviour Blocking	318
15.8 Attacks	319
15.8.1 Hoax	319
15.8.2 Back-door Attack	320
15.8.3 Brute Force Attack	320
15.8.4 Dictionary Attack	321
15.8.5 Spoofing Attack	321
15.8.6 Denial-of-service Attack (DoS attack)	323
15.8.7 Distributed Denial-of-service Attack	323
15.8.8 Man-in-the-middle Attack	324
15.8.9 Spam	324
15.8.10 E-mail Bombing and Spamming	325
15.8.11 Sniffer	325
15.8.12 Timing Attack	326
Summary	327
Exercises	328
16. Firewall	329-345
16.1 Introduction	329
16.1.1 Characteristics of a Firewall	330
16.2 Packet Filters	330
16.3 Application Level Gateways	331
16.3.1 DMZ (Demilitarized Zone)	332
16.3.2 Security Monitors for Firewalls and Perimeter (Proxy Host) Network	333
16.4 Circuit Level Gateways	333
16.4.1 Benefits of a Firewall	334
16.4.2 Limitations of a Firewall	334
16.5 Firewall Architectures	334
16.5.1 Dual-Homed Host Architecture	335
16.5.2 Screened Host Architecture	336
16.5.3 Screened Subnet Architecture	337
16.6 Trusted System	341
16.6.1 Trusted Systems in Policy Analysis	341
16.7 Access Control	342
16.7.1 Objectives of Access Control	342
16.7.2 Types of Access Control	343
16.7.3 Access Control Matrix	343
Summary	344
Exercises	345

17. Computer Forensics	346–359
17.1 Introduction 346	
17.2 Computer Forensics Investigations 349	
12.2.1 Types of Data 350	
17.3 Areas of Application of Computer Forensics 350	
17.3.1 Public Sector 350	
17.3.2 Private Sector 350	
17.4 Understanding the Suspects 351	
17.4.1 Electronic Evidence Considerations 351	
17.4.2 E-mail Review 355	
17.5 Examples of Computer Forensic 356	
17.6 Free Space and Slack Space 357	
17.7 Weaknesses 357	
Summary 358	
Exercises 359	
Bibliography.....	361
Index.....	363–367

Preface

Cryptography and information security has moved from the confines of academia to all users of computer all over the world. Computer security is a science as well as an art. It is an art because no system can be considered secure without an examination of how it is to be used. It is a science because its theory is based on mathematical constructions, analyses, and proofs. It has increasingly become obvious to everybody that something needs to be done in order to secure our network as well as personal computers. This field of security is a challenging field as technology changes everyday. Thus, there is a need to secure computers and networks from the hackers by developing new algorithms. This book introduces the different areas of security in simple and clear terms. In this text, I have tried to put together the basics of computer security in a compact and concise manner.

The book discusses the importance of using information and the security of that information. The aim of this book is to fulfil the need for a quality textbook on cryptography and information security for the students of Information Technology, Computer Science and Engineering, and Master of Computer Applications. As the government is concentrating more on infrastructure security and to create awareness about information security among the people, there is a clear need to include practitioners and students of other disciplines. This book serves their purpose as well. It also contains advanced topics such as computer forensics.

The book is divided into 17 chapters and attempts to introduce the students to the essentials of cryptography and information security.

Chapter 1 discusses security, its importance, elements of security and the operational model of network security. Various encryption methods, their comparisons, and the cipher techniques used in the cryptography and information security are described in Chapter 2. Chapter 3 deals with different ciphers, DES, 3DES, side channel attacks, differential cryptanalysis, and linear cryptanalysis. Solved problems on DES are also given. Chapter 4 describes the RijnDael, and comparison of AES with DES and 3DES. Chapter 5 dwells on various encryption algorithms such as Blowfish, RC4, RC5, RC6 and IDEA.

Chapter 6 concentrates on the mathematical background required for cryptography and information security. It describes prime number, Fermat's theorem, Euler's theorem, primality test, Chinese remainder theorem, and discrete logarithms. Chapter 7 provides an introduction to public key encryption, the RSA algorithm, and timing attacks, besides solved problems on RSA. Chapter 8 focuses on key distribution, the Diffie-Hellman key exchange, elliptic curve and zero knowledge proof systems. Chapter 9 describes the authentication methods, message digest such as MD4, MD5, RIPEMD, SHA and Kerberos, X.509 authentication service. Digital signatures, algorithms, standards and authentication protocols are taken up in Chapter 10.

Chapter 11 introduces the reader to electronic mail security, pretty good privacy (PGP), S/MIME, MIME, and gives a comparison of PGP and S/MIME. Chapter 12 explains IP Security architecture, IPsec, IPv4, IPv6, the authentication header (AH) protocol, the encapsulating security payload (ESP) protocol, the ISAKMP protocol, the OAKLEY key determination protocol, and the key exchange protocol. Secure socket layer, SSL session and connection, the SSL record protocol, secure electronic transactions are explained in Chapter 13.

Chapter 14 describes intrusion detection system, anomaly detection systems, misuse detection system, rule-based intrusion detection, distributed intrusion detection, base-rate fallacy, and password management best practices. Different malicious software is discussed in Chapter 15. Firewall, types of firewall, firewall architecture, and trusted system are explained in Chapter 16. Chapter 17 discusses computer forensics, computer forensics investigations, the areas in which computer forensics are applied, and its drawbacks.

I express my heartfelt gratitude to my wife, Archana, as always, for her positive attitude, full support, and encouragement in my writing endeavours. Thanks to my lovely daughter Samiksha for her continuous support and patience throughout the writing process.

I also wish to thank the editorial and production staff of PHI Learning for their careful processing of the manuscript.

I would like to thank my colleagues and students for their feedback without which it would not have been possible for me to complete this book. Many people have contributed directly or indirectly to this book in a variety of ways. I thank each of them.

Any constructive criticism for the improvement of the book is most welcome.

V.K. Pachghare

Chapter 1

Introduction

1.1 SECURITY

The word ‘security’ signifies the quality or state of being secure, i.e. to be free from any danger. It means to be protected from adversaries who would do harm, intentionally or unintentionally. Security not only protects your network but also allows the authorized people to access your network. It allows the people to work together. To protect the operation of any organization, the following security layers are needed:

- *Physical security*: This protects the physical objects such as hard disk, Random Access Memory (RAM) from unauthorized users.
- *Personal security*: This ensures the protection of the individual or a group of individuals in the organization.
- *Operation security*: This is meant to protect the details of a particular project.

Today, information security is an emerging field as the use of computer is increasing in all walks of life, and the number of security incidents is steadily climbing. Information security, which is also sometimes called *Information Systems Security*, deals with several different “trust” aspects of information and its protection from unauthorized persons.

Information security is not confined to computer systems, nor to information in an electronic or machine-readable form. It applies to all aspects of safeguarding or protecting information or data, in whatever form or media.

It is necessary to protect the information systems against unauthorized access or modification such as deletion or addition of some part into the information, whether in storage, processing or transit. It is also necessary to protect the information system against the denial of service to authorized users or the provision of service to unauthorized users, including those measures necessary to detect, document, and counter such threats.

Most of the definitions of information security tend to focus on specific usages or particular media, for example, “protect electronic data such as email account, banking information from unauthorized use”. In fact, it is a common misconception, or misunderstanding, that information security is synonymous with computer security—in any of its guises:

- Computer and network security
- Information technology (IT) security
- Information systems security
- Information and communications technology (ICT) security.

Each of these has a different emphasis, but the common concern is the security of information or data in some form (electronic in these cases) from unauthorized users. Therefore, all the above categories are subsets of information security. Conversely, information security covers not just information but all infrastructure that facilitates its use—processes, systems, services, technology, etc. including computers, voice and data networks.

It is an important point that information security is not, inherently and necessarily, hermetic, or watertight, or perfectible. No one can ever eradicate all risks of improper or capricious use of any information. The level of information security sought in any particular situation should be commensurate with the value of the information and the loss, financial or otherwise, that might accrue from improper use—disclosure, degradation, denial, or whatever.

1.2 ELEMENTS OF INFORMATION SECURITY

The three widely used elements of information security are confidentiality, integrity and availability. Each of them is discussed here.

1.2.1 Confidentiality

One of the most important elements of information security is confidentiality. It ensures that information is accessible only to those individuals who are authorized. No unauthorized individual or group should ever be able to view data they are not entitled to. But the implementation of this is very difficult as it is not defined which party or people or systems are authorized to use the data and which are not.

Confidentiality also refers to an ethical principle associated with several professions, e.g., medicine, law, religion, journalism. In ethics, (and in some places, in law), certain types of communication between a person and one of these professionals are “privileged” and may not be discussed or divulged to third parties.

1.2.2 Integrity

In cryptography and information security, integrity refers to the validity of data. Integrity means that assets can be modified only by authorized parties and only in an authorized way. It gives the assurance that whatever the data received are exactly as

sent by authorized parties, i.e. no modification including writing, changing status, deleting, and creating anything new in the data.

1.2.3 Availability

Today, the most important aspect of information security is availability. It is the degree to which a system, subsystem, or equipment is accessible and usable upon demand by an authorized party or system according to the specification required for the system. Simply put, availability is the proportion of time a system is in a functioning condition. For example, if some person or system has legitimate access to a particular set of objects, that access should not be prevented.

Historically, up to about 1990, confidentiality was the most important element of information security, followed by integrity, and then availability. By 2001, this scenario had changed and the use and expectation patterns moved availability to the top position of this priority list. The first goal of modern information security, in effect, is to ensure that systems are predictably dependable in the face of all sorts of malice, particularly in the face of denial of service attacks.

The Underlying Technical Models for Information Technology Security added assurance as an essential element. “Without this assurance the other objectives are not met.” Assurance is the basis for confidence that the security measures, both technical and operational, work as intended to protect the system and the information in processes and that the other four security objectives, such as integrity, availability, confidentiality, and accountability, are adequately met by a specific implementation.

1.3 SECURITY POLICY

Computer security, one of the fields of computer science, is concerned with the control of risks related to computer use. The means traditionally taken to realize this objective is to attempt to create a secure computing platform, so designed that users or programs can perform only those actions that have been allowed to them. This involves specifying and implementing a security policy. The actions in question can be reduced to operations of access, modification and deletion. Computer security can be seen as a subfield of security engineering, which looks at broader security issues in addition to computer security.

In a secure system the authorized users of that system are still able to do what they should be able to do or allowed to do. While one might be able to secure a computer beyond misuse by an unauthorized user using various measures.

It is important to distinguish the techniques used to increase a system’s security from the issues of that system’s security status. In particular, systems which have certain fundamental flaws in their security designs cannot be made secure without compromising their usability. Consequently, most computer systems cannot be made secure even after the application of extensive “computer security” measures. Furthermore, if they are made secure, it is often to the detriment of usability.

There are two different approaches to security in computing. The first approach mainly focuses on external threats, and generally treats the computer system itself as

a trusted system. The second approach focuses on the computer system itself as an untrusted system, and redesigns it to make it more secure in a number of ways.

1.4 SECURITY TECHNIQUES

There are two techniques used for creating secure systems: cryptographic techniques and strong authentication techniques. Cryptographic techniques can be used to defend data in transit between systems, reducing the probability that data exchanged between systems can be intercepted or modified. Strong authentication techniques can be used to ensure that communication end-points are who they say they are.

Secure cryptoprocessors can be used to leverage physical security techniques into protecting the security of the computer system.

1. *Chain of trust techniques*: This technique can be used to ensure that all software loaded has been certified as authentic by the system's designers.
2. *Mandatory access control*: This technique can be used to ensure that privileged access is withdrawn when privileges are revoked. For example, deleting a user account should also stop any processes that are running with that user's privileges.
3. *Capability and access control list techniques*: This technique allows the separation of privilege and mandatory access control.
4. *Capability to detect unpatched known flaws*: In a production system when an application provides no way to patch already known security flaws, do not use it or use another one (at least until the fix is available). The worms use the publicly known flaws for their entry and automatically break into a system and then spread to other systems connected to it. The website *Secunia* provides a search tool for unpatched known flaws.
5. *Backup*: An important way of securing your information is backup. It helps us to save another copy of all your important computer files kept in another location. These files are kept on hard disks, CD, pen drive and floppy. Backups can be kept in a multitude of locations, such as a fireproof, waterproof, and heatproof safe, or in a separate, offsite location in which the original files are contained. Some individuals and companies also keep their backups in safe deposit boxes inside the vaults of banks. There is another option, which involves using one of the companies on the Internet that backup files for both business and individuals.

Backups are also important for reasons other than security. Natural disasters, such as earthquakes, hurricanes, or tornadoes, may strike the building where the computer is located. The building can be on fire, or an explosion may occur. There needs to be a recent backup at an alternate secure location, in the case of such kind of disaster. The backup needs to be moved between the geographic sites in a secure manner, so as to prevent it from being stolen.

6. *Anti-virus software*: It consists of programs that attempt to identify, thwart and eliminate computer viruses and other malicious software (malware). It protects the computer as well as information on the system from viruses.
7. *Firewalls*: These are the systems which help to protect computers and computer networks from attack and subsequent intrusion by restricting the network traffic which can pass through them, based on a set of rules defined by the system administrator.
8. *Access authorization*: It helps to restrict access to a computer to an individual or a group of users through the use of authentication systems. These systems can protect either the whole computer through an interactive logon screen or individual services, such as an FTP server. There are many methods for identifying and authenticating users, such as passwords, identification cards, and, more recently, smart cards and biometric systems (fingerprint or iris).
9. *Encryption*: It is used to protect your message from others. It can be done in several ways by switching the characters around, replacing characters with others, and even removing characters from the message. These have to be used in combination to make the encryption secure enough, that is to say, sufficiently difficult to crack. Symmetric and asymmetric encryptions are the methods used for encryption. They allow, for example, any one to write a message for a list of recipients, and only those recipients will be able to read that message.
10. *Intrusion-detection systems (IDS)*: It is a type of security management system for computers and networks. An IDS gathers and analyzes information from various areas within a computer or a network to identify possible security breaches, which include both intrusions (attacks from outside the organization) and misuse (attacks from within the organization).
11. *Social engineering awareness*: Keeping yourself and your employees aware of the dangers of social engineering and/or having a policy in place to prevent social engineering can reduce successful breaches of your network and servers.

1.5 STEPS FOR BETTER SECURITY

For better security, five steps are needed. They are as follows:

Step 1 Assets: First decide what is to be protected. Identify the assets, i.e. the important information that will be protected by security measures.

Step 2 Risks: The second step is the risks in your project. Identify the threat, attacks, vulnerabilities and risks after the assets to be protected have been identified.

Step 3 Protections: Find out the solution for the protection of the assets.

- Step 4 Tools:* Select the appropriate tools for the protection of the assets.
- Step 5 Priorities:* Decide the order of the tools and techniques for the protection of the assets.

1.6 CATEGORY OF COMPUTER SECURITY

Computer security can be categorized into the following:

Authentication is the process of establishing or confirming a proof of identities, that is, that claims made by or about the thing are true. Authenticating an object means confirming its provenance, whereas authenticating a person often consists of verifying his identity. Authentication depends upon various factors.

In computer security, authentication is the process of attempting to verify the digital identity of the sender of a piece of information. The sender being authenticated may be a person using a computer, a computer itself or a computer program.

Cryptography is the study of ways to convert information from its normal, comprehensible form into an obscured guise, unreadable without special knowledge — the encryption. In the past, cryptography helped ensure secrecy in important communications, such as those of spies, military leaders, and diplomats. In recent decades, the field of cryptography has expanded its area. Examples include schemes such as digital signatures and digital cash, digital rights management for intellectual property protection, and securing electronic commerce. Cryptography is now often built into the infrastructure for computing and telecommunications; users may not even be aware of its presence.

Data security is the means of ensuring that data is kept safe from corruption and access to it is suitably controlled. Thus, data security helps ensure privacy. It also helps in protecting personal data.

Formal methods are mathematical approaches to software and hardware computer-based system development, from requirements, specification and design to programming and implementation. They form an important theoretical underpinning for software engineering, especially where safety or security is involved. Formal methods are a useful adjunct to software testing since they help to avoid errors and can also give a framework for testing.

Identity management is the management of information which represents real-life identified items such as users, devices, services, etc.

Internet privacy signifies privacy over the media of the Internet, i.e. the ability to control what information one reveals about oneself over the Internet, and to control who can access that information. Many people use the term to mean universal Internet privacy: every user of the Internet possessing Internet privacy.

Computer security models refer to the underlying computer architectures, protection mechanisms, distributed computing environment security issues, and formal models that provide the framework for information systems security policy.

Network security consists of the provisions made in an underlying computer network infrastructure, policies adopted by the network administrator to protect the network and resources from unauthorized access and the effectiveness (or lack) of these measures combined together. Network security starts from the authentication of the user.

Computer security procedures include strategies, guidelines, policies, standards, specifications, regulations and laws.

Security exploit is related to computer security vulnerabilities and their exploits. It includes computer hacking, cryptographic attacks, denial of service attacks, Malware, etc.

Security software is a generic term referring to any computer program or library whose purpose is to secure a computer system or computer network. Examples are anti-virus software, Firewall software.

1.7 THE OPERATIONAL MODEL OF NETWORK SECURITY

It was assumed earlier that if we prevented somebody from accessing our computer systems and networks, we obtained security. The basic idea of this is true, but this is not the fact today. It fails to acknowledge the realities of the networked environment of which our system is a part. So, protection does not mean prevention. The security aspect comes into play when two parties want to communicate with each other. We should protect our data from any third party or unauthorized person or opponent by providing security.

We follow some steps for secure transmission of the data such as:

- Use of a secure algorithm for transmission of the message
- Use some secret information, keywords with the algorithm
- Use some innovative ideas for the distribution and sharing of the secret information, keywords.
- Specify a protocol to be used for transmission of the message by the two parties.

Figure 1.1 shows the operational model of network security. From Figure 1.1, suppose sender A wants to communicate with B. He writes some message and hand over it to a person to give it to B without sealing it. Before the message is delivered to B, any third person can read it. So to avoid this, A must seal the message in an envelope and hand over it to a person to give it to B. Now only B can break the seal and read the message. That is, security is provided to the message. The same process applies for the network security model. Here, the key indicates security mechanism. It may be a password to the document, so that only authorized persons who know the password can access the message. Unauthorized persons cannot read the message until they know the password. In this book, the various techniques are explained for protecting data from unauthorized persons.

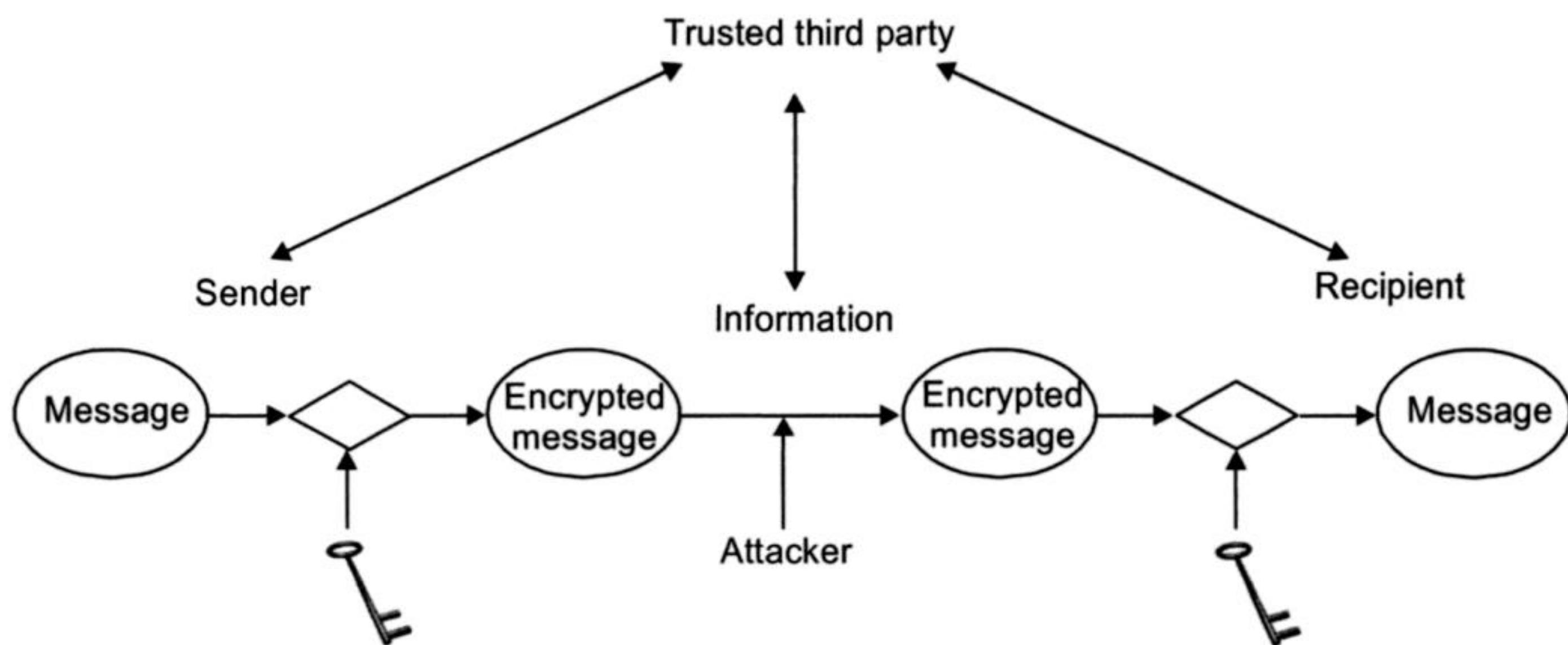


Figure 1.1 Operational model of network security.

1.8 BASIC NETWORK SECURITY TERMINOLOGY

Cryptography: It is the science of using mathematics to encrypt and decrypt data. Cryptography is the art of secret writing. It enables you to store information or transmit it across insecure networks, so that it cannot be read by anyone except the intended recipient.

Hacking: Most frequently used term in the media. A hacker was once considered an individual who understood the technical aspects of computer operating systems and networks. A hacker is a person who creates and modifies computer software and computer hardware, including computer programming, administration, and security-related items. The term usually bears strong connotations, but may be either favourable or denigrating depending on cultural context.

In computer programming, a hacker is a programmer who hacks or reaches a goal by employing a series of modifications to exploit or extend existing code or resources. For some, “hacker” has a negative connotation and refers to a person who “hacks” or uses kludges to accomplish programming tasks that are ugly, inelegant, and inefficient. This negative form of the noun “hack” is even used among users of the positive sense of “hacker”.

In computer security, a hacker is a person who is a master in working with the security mechanisms for computer and network systems. True hackers are honest and careful not to harm anyone. While including those who endeavour to strengthen such mechanisms, it is more often used, especially in the mass media, to refer to those who seek access.

In other technical fields, a hacker means a person who makes things work beyond perceived limits through his own technical skill, such as a hardware hacker, or reality hacker.

In hacker culture, a hacker is a person who has attained a certain social status and is recognized among members of the culture for commitment to the culture’s values and a certain amount of technical knowledge.

Encryption: The translation of data (plaintext) into a secret code (ciphertext) is called encryption. Encryption is the most secure way to achieve data security.

There are two main types of encryption: asymmetric encryption (also called public-key encryption) and symmetric encryption.

Decryption: It is the process of decoding data (ciphertext) that has been encrypted into a secret format. Decryption requires a secret key or password.

Governance: It is the processes and systems by which an organization or society operates.

Access control: It includes authentication, authorization and audit. Describes additional measures such as physical devices, including biometric scans and metal locks, hidden paths, digital signatures, PIN, encryption, social barriers, and monitoring by humans and automated systems. It helps in protecting the system from the unauthorized users.

Risk assessment: This is a step in the risk management process. It helps us to know the risk for a particular project.

Return on information security investment (ROISI): It is a methodology that attempts to find the viability of an information security program. It provides us the idea about how much security is to be provided according to the importance of the project. The methodology strikes a balance between underspending and overspending, the effort on the project.

Non-repudiation: It means that a contract, especially one agreed to via the Internet, cannot later be denied by one of the parties involved. In regard to digital security, non-repudiation means the fact that the sender and the recipient were, in fact, the parties who claimed to send or receive the message, can be verified. In other words, non-repudiation of origin proves that the data has been sent, and non-repudiation of delivery proves it has been received.

Authorization: In information security or computer security, authorization is a part of the operating system that protects computer resources by allowing only those resources to be used by resource consumers that have been granted authority to use them. Resources include individual files or items data, program, computer devices, and functionality provided by computer applications. Resource consumers are computer users such as administrator who can use all the resources of the computer but he can create new user on his system on the name of guest and allow using only limited resources. In this case, the guest user can use the system but only those resources that are allowed to him by the administrator. Other users can use the system or any of its resources.

Computer security audit: It is a process that can verify certain standards that have been met.

Alerting assurance and reliability: Assurance services are also called *Independent Professional Services*. It improves information quality or its context. Such services could include assessments of Internet security and quality of health facilities. In general, reliability (systemic definition) is the ability of a system to perform and maintain its functions in routine circumstances.

Business Continuity Planning (BCP): This is a methodology used to create a plan for an organization regarding how it will resume partially or completely interrupted critical function(s) within a predetermined time after a disaster or disruption.

Communications Security (COMSEC): This involves measures and controls taken to deny unauthorized persons information derived from telecommunications and to ensure the authenticity of such telecommunications. Communications security includes cryptosecurity, transmission security, emission security, traffic-flow security, and physical security of COMSEC material.

Cryptography and cryptanalysis: These are important tools in assuring confidentiality, integrity, and source identification. It is always assumed that the key(s), may be public or private involved in cryptography have not been misused or compromised, and that the encryption, decryption algorithms employed have been well chosen and properly used.

SUMMARY

Hopefully, you are now convinced of the importance of security. It indicates the quality or state of being secure. Security is good for you and your data. It is also necessary to protect the information system against the denial of service to authorized users or the provision of service to unauthorized users, including those measures necessary to detect, document, and counter such threats. Security means doing what you can to protect, or at least to avoid endangering, the network and computers used by yourself and others. The growth of e-commerce and the pervasive personal and business uses of the Internet have created a demand for security. In this chapter, you learnt about the security services which are more important to keep our information secure.

EXERCISES

1. List the elements of information security and explain each in brief.
2. Explain confidentiality, integrity, and availability.
3. Explain the following terms with respect to Information Security:
(a) Access control, (b) Risk assessment, (c) Non-repudiation, (d) Authorization,
(e) Confidentiality, (f) Integrity, (g) Encryption, (h) Decryption.
4. Explain the operational security model with block diagram.
5. What is authentication?

Chapter

2

Data Encryption Techniques

2.1 INTRODUCTION

Often, there is a need to protect information from ‘prying eyes’. In the electronic age, the information that could otherwise benefit or educate a group or individuals can also be used against such groups or individuals. Industrial espionage among highly competitive businesses often requires that extensive security measures should be put into place. And those who wish to exercise their personal freedom, outside the oppressive nature of governments, may also wish to encrypt certain information to avoid suffering the penalties of going against the wishes of persons who attempt to control.

Encryption is the process of encoding information (plaintext) to make it unreadable without special knowledge. While encoding, the meaning of the message is not obvious. Decryption is the reverse process. The process of decoding or transforming an encrypted message (ciphertext) back to its readable and original form (plaintext) is called *decryption*. In other words, encryption helps us to hide the meaning of the original message so that we can send it safely, and decryption helps us reveal the original message from the secret message. In encryption, we use various techniques to encode the message whereas in decryption, the prior knowledge of key or password is required to decode the message. Thus, encryption and decryption help in secure transmission of the message and in protecting the message from unauthorized persons.

Figure 2.1 explains encryption and decryption. A system for encryption and decryption is called a *cryptosystem*. The techniques used for enciphering constitute the

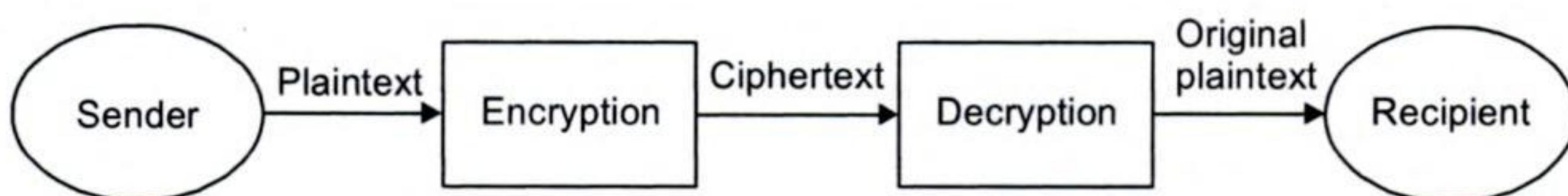


Figure 2.1 Encryption and decryption.

area of study known as *cryptography*. The techniques used for deciphering a message without any knowledge of the enciphering details fall into the area of *cryptanalysis*, also called *Breaking the code*. The areas of cryptography and cryptanalysis together are called *cryptology*.

2.2 ENCRYPTION METHODS

The methods of data encryption and decryption are relatively straightforward, and can be mastered easily. We divide encryption methods into two parts:

1. Symmetric encryption
2. Asymmetric encryption

2.2.1 Symmetric Encryption

Symmetric encryption is also referred as *conventional encryption*. For example, A and B agreed on an encryption method and a shared key. A uses the key and the encryption method to encrypt (or encipher) a message and sends it to B. B uses the same key and the related decryption method to decrypt (or decipher) the message as shown in Figure 2.2. Symmetric encryption is similar to the process followed by most people to make their home safe. A person purchases a lock of some company to close

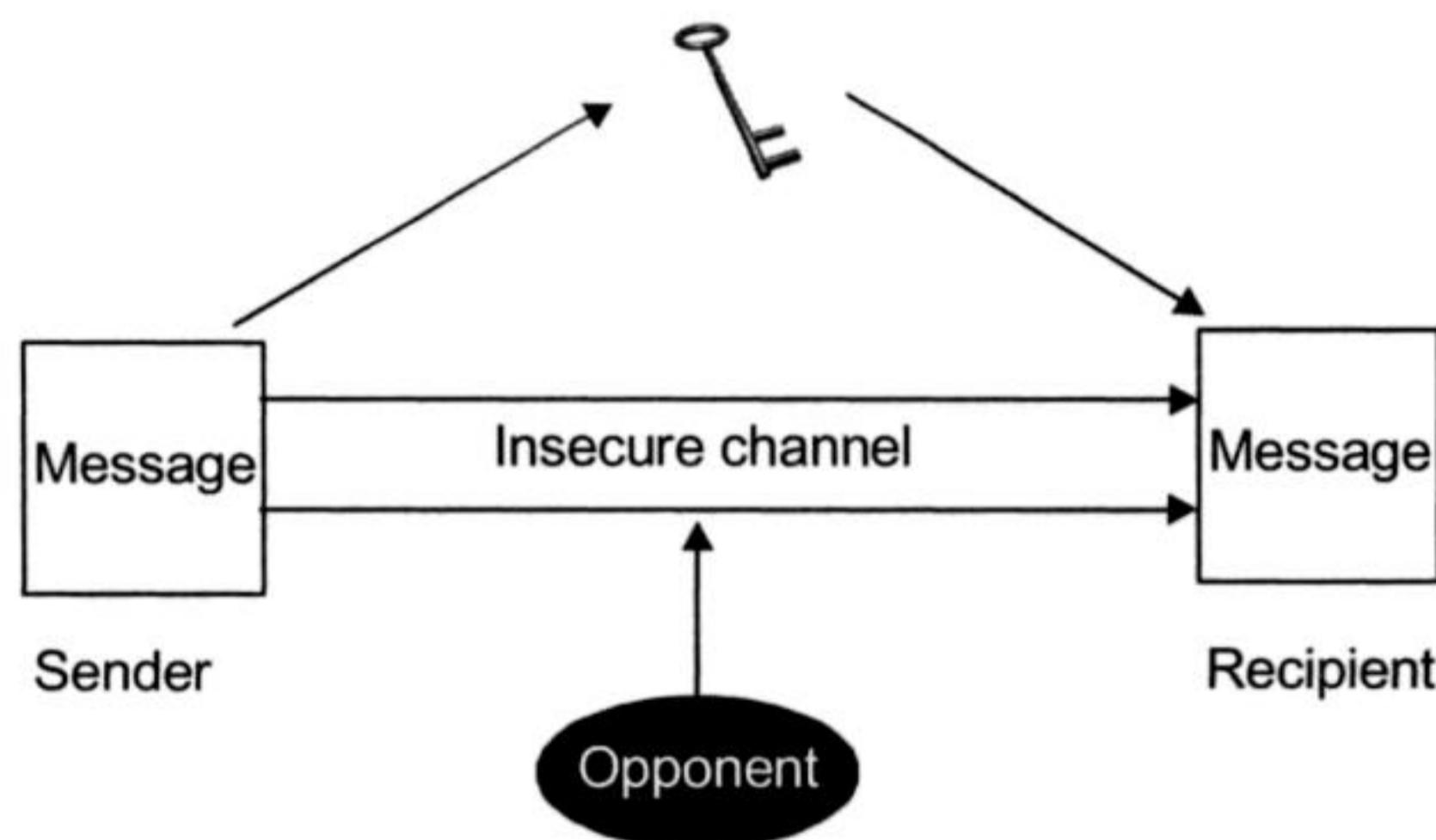


Figure 2.2 Symmetric encryption and decryption.

his door. The same key is required to open and lock the door. Another key from the same company cannot open or close the lock. Similarly, in symmetric encryption the same key is required for encryption as well as decryption. A few well-examined encryption algorithms that everyone could use just like the model of the lock may be the same, but the keys are different.

Figure 2.3 shows the various components of the symmetric encryption and decryption. The components of symmetric encryption are the following:

1. **Plaintext:** This is the original message written or created by the sender used as input for the encryption algorithm.

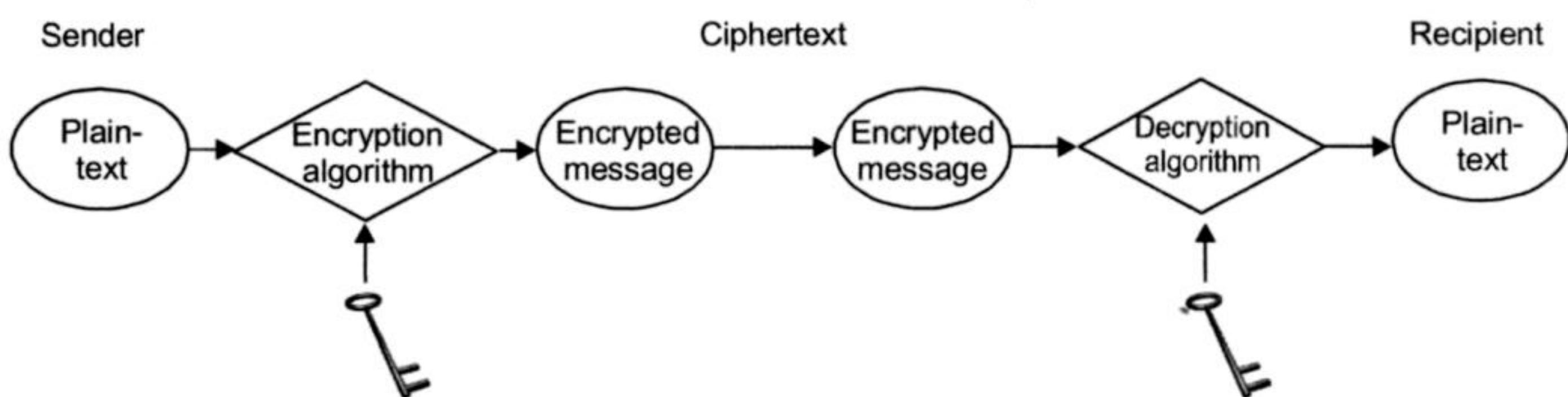


Figure 2.3 Components of symmetric encryption and decryption.

2. Encryption algorithm: Here we have to use various encryption algorithms to encrypt the plaintext. This helps us to convert the plaintext into ciphertext.

An encryption algorithm is called *breakable* when, given enough time and data, an analyst can determine the algorithm. However, if the algorithm is theoretically breakable, it is in fact, impractical to try to break it. Practically, however, things are different. Estimates of breakability are based on current technology. Those algorithms which require thousands of years to break may break within a day due to advance technology. Technology evolves very fast. Most encryption algorithms are mathematical in nature or can be explained and studied with mathematics. Text symbols are coded with numbers and encryption operates on the numerical representation of the symbols (ex. ASCII codes).

3. Key: Key is used as the input to the encryption algorithm. The key is the same for encryption and decryption. For every new message transformation, we can use new key to produce more security.

4. Ciphertext: This is the output of the plaintext after encryption. This depends on the key and the plaintext. Two different keys will produce two different ciphertexts. The ciphertext is a set or random stream of data.

5. Decryption algorithm: The decryption algorithm runs in reverse of the encryption algorithm. The input of this algorithm is the ciphertext and the key. The key for decryption is the same as that used for encryption.

2.3 CRYPTOGRAPHY

Cryptography is the practice of using encryption to conceal text.

Cryptographic systems are characterized as:

- *The type of operations used for transforming plaintext to ciphertext:* All the encryption algorithms are based on substitution in which one element is replaced for another, and transposition in which the order of the elements is rearranged. Most systems involve multiple stages of substitutions and transpositions.

- *The number of key used:* In symmetric encryption, only one key is used for encryption and decryption. It is referred as *secret key*, single key or conventional encryption. In asymmetric encryption, two keys are used. It is also referred as two key, or public key encryption.
- *Ways of processing the plaintext:* There are two ways to process the plaintext: viz. block cipher, and stream cipher. In the block cipher, the input plaintext is divided into blocks and each block is processed at a time. The result of block cipher is one block for each block of input plaintext. In stream cipher, we process the input elements continuously one by one. The result of steam cipher is one element for each element of input plaintext.

2.4 CRYPTANALYSIS

Cryptanalysis is the study of methods for obtaining the meaning of encrypted information, without access to the secret information which is normally required to do so. Typically, this involves finding the secret key. In non-technical language, this is the practice of trying to break any ciphertext message to obtain the original message called *plaintext*. The person who attempts to break the security is called the *cryptanalyst*.

The cryptanalyst can break encrypted message (ciphertext) by any or all of the following ways.

- Breaking the message
- Recognizing patterns in encrypted messages in order to be able to break subsequent ones
- Concluding some meaning from the ciphertext without breaking it
- Finding general weaknesses in an encryption algorithm
- Deducing the key in order to break subsequent messages easily

The cryptanalyst tries every possible key on a piece of ciphertext until an intelligible translation into plaintext is obtained. Therefore, to make the secure message transformation, we need a strong encryption algorithm so that the cryptanalyst or the opponent who knows the algorithm and has access to one or more ciphertexts is unable to decrypt it, and also unable to find out the key. We also need to keep the key secure. Key may be generated using random number generation and for every new message transformation, use new key.

There is no need to keep the symmetric encryption algorithm secret. Only keep the key secure.

We said the algorithm is secure if the cost of breaking the ciphertext exceeds the value of the encrypted information, and also the time required to break the ciphertext exceeds the useful lifetime of the information. If an encryption algorithm fulfils these two criteria, then it is computationally secure.

Throughout this book, we have used the convention that plaintext is written in lower case letters and ciphertext in uppercase letters. As most of the encryption algorithms are based on mathematical transformation, they can be studied more easily in the mathematical form.

The following example shows the ciphertext for the plaintext.

Plaintext	a	b	c	d	e	f	g	h	i	j	k	l	m
Ciphertext	5	6	7	8	9	10	11	12	13	14	15	16	17
Plaintext	n	o	p	q	r	s	t	u	v	w	x	y	z
Ciphertext	18	19	20	21	22	23	24	25	26	1	2	3	4

Thus, the letter 'A' is represented by 5, 'B' by 6, and so on. This representation allows us to convert the letters into the numbers. In Section 2.5, we discuss various ciphers.

2.5 SUBSTITUTION CIPHERS

The two building blocks of all encryption techniques are: substitution ciphers and transposition ciphers.

The technique in which the elements of plaintext are replaced by other elements or by numbers or symbols is called *substitution ciphers*. Substitution ciphers are called *monoalphabetic ciphers* (simple substitution). If the plaintext is viewed as a sequence of bits, then substitution involves replacing plaintext bit patterns with ciphertext bit patterns.

Here, we discuss different types of substitution ciphers. *Monoalphabetic Ciphers*, *Playfair cipher*.

2.5.1 The Caesar Cipher

The simplest of all substitution ciphers is the one in which the 'cipher letters' are generated by shifting 'plaintext letters' by the same distance. Julius Caesar was the person who first proposed what is known as *Caesar cipher*. He used a method in which each letter was translated to a letter a fixed number of places after the position of that letter in the alphabet. If the language of the plaintext is unknown, then the plaintext output may not be recognizable and the input may be abbreviated or compressed in such a way that recognition of the same is difficult. This makes the algorithm more secure. A secure encryption should not allow an interceptor to use a small piece of the ciphertext to predict the entire pattern of the encryption. Julius Caesar used a shift of 3, so the plaintext letter PT_i is enciphered as ciphertext letter CT_i such as

$$CT_i = E(PT_i) = PT_i + 3$$

A full translation chart of the Caesar cipher is given below.

Plaintext	a	b	c	d	e	f	g	h	i	j	k	l	m
Ciphertext	D	E	F	G	H	I	J	K	L	M	N	O	P
Plaintext	n	o	p	q	r	s	t	u	v	w	x	y	z
Ciphertext	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

Using this encryption, the message **work patiently** would be encoded as

Plaintext	w	o	r	k	p	a	t	i	e	n	t	l	y
Ciphertext	Z	R	U	N	S	D	W	L	H	Q	W	O	B

Advantages

- Easy to implement.
- Cipher is quite simple.

Disadvantages

- Brute force cryptanalysis is easily performed.
- Its obvious pattern is the major drawback.
- With only 25 possible keys, this algorithm is not secure.

2.5.2 Monoalphabetic Ciphers

The monoalphabetic cipher (often referred to as a *cryptogram*) uses a KEY, which is the rearrangement of the letters of the alphabet. These different letters are then substituted for the letters in the plaintext to create a ciphertext. That KEY is needed to decipher the secret message. The cipher line can be a permutation of the 26 alphabetic characters, and then there are $26!$ Or greater than 4×10^{26} possible keys, which help eliminate the brute force techniques for cryptanalysis. It is known as *monoalphabetic substitution cipher* as a single cipher alphabet is used per message. For example, we can use the permutation of the numbers 1 to 10 in many ways like Permutation (P) $P_1 = 2, 4, 6, 8, 10, 1, 3, 5, 7, 9$ or $P_2 = 9, 7, 5, 3, 1, 2, 4, 6, 8, 10$ and a permutation function can be written as $P_1(2) = 4$, meaning that the letter in position 2 is to be replaced by the fourth letter. We can also use a key, i.e. a word that controls the permutation. Suppose if the key is 'ANOTHER', write it under the first letters of the alphabet.

Plaintext	a	b	c	d	e	f	g	h	i	j	k	l	m
Ciphertext	A	N	O	T	H	E	R						
Plaintext	n	o	p	q	r	s	t	u	v	w	x	y	z
Ciphertext													

Then, fill the remaining letters of the alphabet in some easy way, after the keyword.

Plaintext	a	b	c	d	e	f	g	h	i	j	k	l	m
Ciphertext	A	N	O	T	H	E	R	B	C	D	F	G	I
Plaintext	n	o	p	q	r	s	t	u	v	w	x	y	z
Ciphertext	J	K	L	M	P	Q	S	U	V	W	X	Y	Z

In this cipher if the key is short, then most of the plaintext letters are only one or two positions off from their ciphertext equivalent. So, key should be used in which the distance is large and less predictable. We take another example. Suppose if the key is 'ASSIGNMENT'.

Plaintext	a	b	c	d	e	f	g	h	i	j	k	l	m
Ciphertext	A	S	I	G	N	M	E	T	B	C	D	F	H
Plaintext	n	o	p	q	r	s	t	u	v	w	x	y	z
Ciphertext	J	K	L	O	P	Q	R	U	V	W	X	Y	Z

As P must match one plaintext letter to exactly one ciphertext letter, duplicate letters in a keyword, such as S and N in ASSIGNMENT, are dropped. Near the end of the alphabet, replacements are rather close and the last six alphabets remain the same, i.e. the Ciphertext for 'u' is U, for 'v' is V, etc.

Cryptanalysis of Monoalphabetic Ciphers

At face value monoalphabetic ciphers require $26!$ decipherments. However, using statistical text analysis (frequency of the appearance of letters in the language), deciphering becomes very easy. Monoalphabetic ciphers are easy to break because they reflect the frequency count of the data of the original alphabet.

2.5.3 Playfair Cipher

Although the Baron Playfair's name is attached to one of the better-known classical ciphers, Playfair's friend, Charles Wheatstone, actually devised the Playfair cipher. After its creation in 1854, the Baron successfully lobbied the British government to adopt the cipher for official use, and thus got his name, and not Wheatstone's, attached to the cipher. It is the best known encryption cipher which treats diagrams in the plaintext as single units and translates these units into ciphertext diagrams. The Playfair is a primitive block cipher.

Preparing the Plaintext

The first step is to prepare the plaintext. To do so, all letters are written in lower case, in pairs, and without punctuation. If j is present all j's are replaced with i's. This particular example contains no j's.

Let us take the example: We live in a world full of beauty.

we	li	ve	in	aw
or	ld	fu	ll	of
be	au	ty		

Next, double letters—if they occur in a pair—must be divided by a X or a Z. For example, 'full' in this example becomes 'fulxl'. This rule for double letters reduces the number of visible patterns in the ciphertext.

Finally, if there are an odd number of letters, an extra letter chosen by the person writing the cipher is added in the end.

In its ready-to-encrypt form, the plaintext becomes:

we	li	ve	in	aw
or	ld	fu	lx	lo
fb	ea	ut	yz	

Preparing the Key

The alphabet square is a five-by-five matrix of letters constructed using a keyword. The key phrase is first written without repeating any letters. The remaining letters of the alphabet are filled in the alphabetic order as shown below:

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

In this case, the keyword is ‘ANOTHER’. The matrix is constructed by filling in the letters of the keyword from left to right and from top to bottom. Then other letters from A to Z are filled in the remaining part of the matrix. Note the absence of a J, and that the some letters of the key had already appeared in the previous seven.

The longer your key, the more secure your message will be—but remember that even the most secure Playfair cipher can be broken easily with the proper tools.

Encryption

Next comes the heart of the Playfair cipher. Any pair of letters must be in the same row, in the same column, or in different rows and columns; no other combinations are possible. The plaintext is encrypted two letters at a time using the following steps:

- Step 1** Each letter in a pair that is on the same row is replaced by the letter to the right. The letter to the right of the rightmost letter is the first letter in the same row—it “wraps” around without going to the next line. With this key, nt becomes OH; yw becomes ZX, and so on.
- Step 2** Similarly, letters in the same column are replaced by the next letter below in the same column, i.e. em becomes FV, tk becomes CS.
- Step 3** When the letters are neither in the same row nor in the same column, then the substitution is based upon their intersection. Of great importance is preserving the order. Start with the first letter, and move across until it is lined up with the second letter; then start with the second, and move up or down until it is lined up with the first. gs becomes KP, ar becomes NE. Remember, first move across (left or right), and then up or down. Finally, perform this transformation for each pair of letters in the modified plaintext and remove the spaces. The ciphertext for the above plaintext is:

VRFKAFGONVNULMLIZIHEIFESHZV

In simple monoalphabetic ciphers, there are only 26 letters but in the Playfair cipher, there are $26 \times 26 = 676$ diagrams. The identification of individual diagrams is more difficult, and also the frequencies of individual letters have greater ranges which provide more security.

Decryption

To decrypt the message, simply reverse the entire process. Break the ciphertext into pairs of letters:

VR	FK	AF	GO	NV
NB	UL	ML	I _Z	IH
EI	FE	SH	ZV	

Write down the alphabet square with the key “ANOTHER”:

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

Transform the pairs of letters in the opposite direction from that used for encryption:

WE	LI	VE	IN	AW
OR	LD	FU	LX	LO
FB	EA	UT	YZ	

This message is now readable, although removing the extra spaces and substitutions for double letters makes it more readable:

We live in a world full of beauty.

Modern ciphers are not restricted to upper case, no punctuation, J-less messages. Any form of data that can be stored on a computer can be encrypted with a modern cipher. Modern block cipher can be run in a mode similar to that of Playfair, where the same block (in Playfair, a pair of letters) always encrypts to the same bit of ciphertext.

The Playfair cipher is relatively easy to break, because it still leaves much of the structure of the original language used for the plaintext.

2.5.4 The Hill Cipher

In classical cryptography, the Hill cipher is a polygraphic substitution cipher based on linear algebra. Invented by Lester S. Hill in 1929, it was the first polygraphic cipher in which it was practical (though barely) to operate on more than three symbols at a time.

Working

Each letter is assigned a digit in base 26: A = 0, B = 1, and so on. A block of n letters is then considered as a vector of n dimensions, and multiplied by a $n \times n$ matrix, modulo 26. The components of the matrix are the key, and should be random provided that the matrix is invertible in \mathbb{Z}_{26}^n (to ensure decryption is possible). Consider the message ‘COE’, and the key below (or “ANOTHERBZ” in letters):

Ciphertext = Key × Plaintext mod 26

$$C = KP \bmod 26$$

$$\begin{array}{ccc} 0 & 13 & 14 \\ K = 19 & 6 & 4 \\ 17 & 1 & 25 \end{array}$$

Since 'C' is 2, 'O' is 14 and 'E' is 4, the message is the vector:

$$\begin{array}{c} 2 \\ P = 14 \\ 4 \end{array}$$

Thus, the enciphered vector is given by:

$$\begin{array}{ccccc} 0 & 13 & 14 & 2 & 238 \\ 19 & 6 & 4 & X & 14 = 138 \bmod 26 \\ 17 & 1 & 25 & & 148 \\ K & & & P & KP \bmod 26 \\ 4 & & & & \\ 8 & & & & \\ 18 & & & & \end{array}$$

which corresponds to a ciphertext of 'EIS'.

Every letter has changed. The Hill cipher has achieved Shannon's diffusion, and an n -dimensional Hill cipher can diffuse fully across n symbols at once.

Decryption

In order to decrypt, we turn the ciphertext back into a vector, then simply multiply by the inverse matrix of the key matrix. (There are standard methods to calculate the inverse matrix.) We find that in \mathbb{Z}_{26}^n the inverse matrix of the one in the previous example is:

$$P = K^{-1} \times C \bmod 26$$

$$K^{-1} = \begin{array}{cccc} 1/6453 & -146 & 311 & 32 \\ & 407 & 238 & -266 \\ & 83 & -211 & 247 \end{array}$$

Taking the previous example ciphertext of 'EIS', we get:

$$K^{-1} = \begin{array}{cccc} 1/6453 & -146 & 311 & 32 & 4 \\ & 407 & 238 & -266 & \times 8 \\ & 83 & -211 & 247 & 18 \end{array} \bmod 26$$

which gets us back to 'ACT', just as we hoped.

One complication that we have overlooked, is that inverse of a matrix does not always exist. There is a straightforward way to find this out, though. If the determinant of the matrix is 0, or has common factors with the modulus (i.e. factors of 2 or 13, in the case of modulus 26), then the matrix cannot be used in the Hill cipher; discard it and try another one. Fortunately, unless the basis has small factors, most matrices will have an inverse. Alas! because 2 is one of the factors of 26, quite a few matrices modulo 26 will not work.

Security

The strength of the Hill cipher is that it completely hides single-letter frequencies. So, it is strong against a ciphertext attack. Unfortunately, the basic Hill cipher is vulnerable to a known-plaintext attack because it is completely linear. An opponent who intercepts n^2 plaintext/ciphertext character pairs can set up a linear system which can (usually) be easily solved; if it happens that this system is indeterminate, it is only necessary to add a few more plaintext/ciphertext pairs. Calculating this solution by standard linear algebra algorithms then takes very little time.

Security could be greatly enhanced by combining with some non-linear step to defeat this attack. The combination of wider and wider weak, linear diffusive steps like a Hill cipher, with non-linear substitution steps, ultimately leads to a substitution-permutation network (e.g. a Feistel cipher).

Key Size

One might think that the key size, in bits, is $n^2 \log_2 26$ or about $4.7n^2$. In fact, it is slightly less than this because not all randomly selected matrices are usable. A slightly less naïve view might guess that $1/2 + 1/26$ of candidate keys would be unusable, reducing the keyspace by about 54%. In fact, determinants are not uniformly distributed, and the key space reduction is closer to 70%. Additionally, it seems to be prudent to avoid too many zeroes in the key matrix since they reduce diffusion. The net effect is that the effective keyspace of a basic Hill cipher is about $4.64n^2 - 1.7$. For a 5×5 Hill cipher, it is about 114 bits. Of course, key search is not the most efficient known attack.

Mechanical Implementation

When operating on 2 symbols at a time, a Hill cipher offers no particular advantage over Playfair cipher, and, in fact, is weaker than that, and slightly more laborious to operate by pencil and paper. As the dimension increases, the cipher rapidly becomes infeasible for a human to operate by hand. But astonishingly, a Hill cipher of dimension 6 was once implemented mechanically! Unfortunately, the gearing arrangements (and thus the key) were fixed for any given machine, so triple encryption was recommended for security: a secret non-linear step, followed by the wide diffusive step from the machine, followed by a third secret non-linear step. Such a combination was actually very powerful in 1929, and indicates that Hill apparently understood the concepts of a meet-in-the-middle attack as well as confusion and diffusion. Unfortunately, his machine did not sell.

2.5.5 Polyalphabetic Ciphers

The main problem with simple substitution ciphers is that they are vulnerable to frequency analysis. Given a sufficiently large ciphertext, it can easily be broken by mapping the frequency of its letters to the known frequencies of, say, English text. Therefore, to make ciphers more secure, cryptographers have long been interested in developing enciphering techniques that are immune to frequency analysis. One of the most common approaches is to suppress the normal frequency data by using more than one alphabet to encrypt the message. Another way to improve on the simple monoalphabetic technique is to use different monoalphabetic substitutions as one proceeds through the plaintext message. The next development in cryptography was polyalphabetic ciphers. The idea behind the polyalphabetic cipher is that a single letter can be encrypted to several different letters instead of just one.

We will study the best known and one of the simplest ciphers, Vigenere cipher, which is a polyalphabetic substitution cipher. In this cipher, the set of related monoalphabetic substitution rules consists of the 26 Caesar ciphers, with shifts of 0 through 25. The keyspace consists of all ordered permutations of the alphabet and we call it a Vigenere square. You can see there are 25 rows that can be used as keys, each numbered with the amount it is shifted.

There are two ways to use Vigenere square to form a polyalphabetic cipher.

1. Cycle through all 25 rows in turn. This would mean every twenty-fifth letter is encrypted with the same key.
2. Create a master-key that specifies in which order the keys (or rows) are to be used. This does not have to include all rows. For example, we could use $k = (5, 2, 16)$ and then cycle through these three keys. This would mean every third letter is encrypted with the same key.

For each single letter, you are only using 1 key and encryption and decryption work as with monoalphabetic ciphers.

The Vigenere cipher, proposed by Blaise de Vigenere from the court of Henry III of France in the sixteenth century, is a polyalphabetic substitution based on Table 2.1

Table 2.1

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	
a	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
b	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	A
c	C	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
d	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
e	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	D
f	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
g	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
h	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
i	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
j	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
k	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
l	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
m	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L

Table 2.1 (contd.)

n	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
o	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
p	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
r	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
s	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
t	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
u	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
v	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
w	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
x	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
z	Z	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

Note that each row of the table corresponds to a Caesar cipher. The first row is a shift of 0; the second is a shift of 1; and the last is a shift of 25.

For example, suppose we wish to encipher the plaintext message:

She is very happy and beautiful girl.

Suppose the keyword used is ‘another’. We begin by writing the keyword, repeated as many times as necessary, above the plaintext message. To derive the ciphertext using the tableau, for each letter in the plaintext, one finds the intersection of the row given by the corresponding keyword letter and the column given by the plaintext letter itself to pick out the ciphertext letter.

Keyword:	anoth	erano	thera	nothe	ranot	heran
Plaintext:	sheis	veryh	appya	ndbea	utifu	lgirl
Ciphertext:	SUSBZ	ZVRLV	TWTPA	ARULE	LTVTN	SKZRY

Decipherment of an encrypted message is equally straightforward. One writes the keyword repeatedly above the message:

Keyword	anoth	erano	thera	nothe	ranot	heran
Ciphertext	SUSBZ	ZVRLV	TWTPA	ARULE	LTVTN	SKZRY
Plaintext	SHEIS	VERYH	APPYA	NDBEA	UTIFU	LGIRL

This time one uses the keyword letter to pick a column of the table and then traces down the column to the row containing the ciphertext letter. The top or index of that row is the plaintext letter.

The strength of the Vigenere cipher against frequency analysis is that there are multiple ciphertext letters for each plaintext letter, one for each unique letter of the keyword. We can see this by examining the above ciphertext. Note that there are 3 e’s in the plaintext message and that they have been encrypted by ‘S,’ ‘V,’ ‘L’, respectively. This successfully masks the frequency characteristics of the English ‘e.’ One way of looking at this is to notice that each letter of our keyword ‘another’ picks out 1 of the 26 possible substitution alphabets given in the Vigenere tableau. Thus, any message

encrypted by a Vigenere cipher is a collection of as many simple substitution ciphers as there are letters in the keyword.

The Vigenere cipher has all the features of a useful field cipher, i.e. easily transportable key and tableau, requires no special apparatus, easy to apply, etc.

EXAMPLES

Method 1: Cycle through all keys in order

When doing this, it is useful to have a printed version of the Vigenere square and use a ruler to make sure you are using the correct key. We will be using key 1 to encrypt the b, key 2 to encrypt the e, key 3 to encrypt the a, and so on.

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
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Plaintext: beautiful

Ciphertext: CGDYYOMCU

Method 2: Using $k = (5, 9, 18, 24, 1)$

We will use key 5 to encrypt the b, key 9 to encrypt the e, key 18 to encrypt the a, and so on.

Plaintext: beautiful

Ciphertext: GNSSUNOMJ

Another way of specifying the key for method 2 is to denote each row by the first letter instead of the number of shifts.

Attacking the Vigenere Cipher

Although the Vigenere cipher seems to defeat frequency analysis, it is not a perfect solution. If you discover the block length (the length of the key), you can still apply frequency analysis given a long enough message. For example, with a 5-character master-key, you apply frequency analysis on characters four letters apart, since they will have been encrypted with the same key.

The Vigenere cipher is also susceptible to the chosen-plaintext attack. This attack occurs when the adversary (who is trying to spy on your message) is able to use your cryptographic system to encrypt, but not to decrypt.

2.5.6 One-time Pad

In cryptography, a one-time pad is a system in which the key is a string of random bits, usually generated by a cryptographically strong pseudo-random number. Generator is used only once to encrypt a message that is then decrypted by the receiver using a matching one-time pad and key. The most significant point here is that once an input ciphertext is used, it is never used again for any another message. Mauborgne suggested the scheme of one-time pad. A one-time pad is a very simple yet completely unbreakable *symmetric* cipher. “Symmetric” means it uses the same key for encryption as well as for decryption. As with all symmetric ciphers, the sender must transmit the

key to the recipient via some secure channel, otherwise the recipient will not be able to decrypt the ciphertext. Messages encrypted with keys based on randomness have the advantage that there is theoretically no way to “break the code” by analyzing a succession of messages. Each encryption is unique and bears no relation to the next encryption so that some pattern can be detected. It produces random output that bears no statistical relationship to the plaintext. With a one-time pad, however, the decrypting party must have access to the same key used to encrypt the message and this raises the problem of how to get the key to the decrypting party safely or how to keep both keys secure. One-time pads have sometimes been used when both the parties started out at the same physical location and then separated, each with knowledge of the keys in the one-time pad. The key used in a one-time pad is called a *secret key* because if it is revealed, the messages encrypted with it can easily be deciphered.

How it Works

Typically, a one-time pad is created by generating a string of characters or numbers that will be at least as long as the message. This string of values is generated in some random fashion—for example, by someone pulling numbered balls out of a lottery machine or by using a computer program with a random number generator. The values are written down on a pad. The pads are given to anyone who is likely to send or receive a message. Typically, a pad may be issued as a collection of keys, one for each day in a month, for example, with one key expiring at the end of each day or as soon as it has been used once.

When a message is to be sent, the sender uses the secret key to encrypt each character, one at a time. If a computer is used, each bit in the character (which is usually eight bits in length) is exclusively “OR’ed” with the corresponding bit in the secret key. (With a one-time pad, the encryption algorithm is simply the XOR operation. Where there is some concern about how truly random the key is, it is sometimes combined with another algorithm such as MD5.) Once the one-time pad is used, it cannot be reused. If it is reused, someone who intercepts multiple messages can begin to compare them for similar coding for words that may possibly occur in both messages.

There are as many bits in the key as in the plaintext. This is the primary drawback of a one-time pad, but it is also the source of its perfect security. It is essential that no portion of the key ever be reused for another encryption.

Benefit

The system uses a key that is a randomly generated string of bits (group of numbers or characters) and is the same length as the plaintext message. So, it is highly secure.

Drawback

The drawback of the one-time pad is that because the secret key is very long and used only once, there is the problem of large number of random key generation, and also the problem of key distribution and protection. Securely transmitting the secret key to decode the message is also a problem. The other issues are of the key distribution and protection, as for every new message a key of equal length is needed by both the sender and the receiver. As traditionally used, one-time pads provide no message



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

information to plan what to search for next. The interpreter then searches for this data in the audit trail. The system proceeds this way, accumulating more and more evidence for an intrusion attempt until a threshold is crossed; at this point, it signals an intrusion attempt.

This is a very clean approach. Because the planner and the interpreter know what they are searching for at each step, the large amounts of noise present in audit data can be filtered, leading to excellent performance improvements. In addition, the system can predict the attacker's next move based on the intrusion model. These predictions can be used to verify an intrusion hypothesis, to take preventive measures, or to determine what data to look for next.

However, there are some critical issues related to this system. Firstly, patterns for intrusion scenarios must be easily recognized. Secondly, patterns must always occur in the behaviour being looked for. And finally, patterns must be distinguishing; they must not be associated with any other normal behaviour.

As data is analyzed, the system makes transitions from one state to another. A transition takes place on some Boolean condition being true (for example, the user opening a file). The approach followed in USTAT is to have state transitions from safe to unsafe states based on known attack patterns. To make this model clearer, we illustrate with an example based almost entirely on an example in Ilgun's thesis.

1. The attacker creates a link starting with “-” (say `-x`) to root's setuid shell script containing the `#!/bin/sh` mechanism.
2. The attacker executes `-x`.

The point of this attack is that whenever a hard link to a file is created, a new inode with the target's original permissions is created. Since invoking a script with the `#!/bin/sh` mechanism invokes a subshell, and further, if the name of the subshell begins with a dash, an interactive shell is created, we see that the attacker has obtained an interactive shell with root privileges. The state diagram for this is shown in Figure 14.3. We see that for the final compromised state to be reached, some conditions have to be fulfilled. If these guard conditions are true, then there is almost certainly an intrusion attempt going on. However, if any of these conditions do not hold, the probability of an intrusive action is considerably decreased. We see that the guard conditions exist to filter the intrusive activities from the non-intrusive ones. Hence, this can serve as a data pruning mechanism as observed in the model based scheme above. Some advantages of this approach are: it can detect cooperative attacks, it can detect attacks that span across multiple user sessions, and it can foresee impending compromise situations based on the present system state and take pre-emptive measures.

However, there are also a few problems with state transition systems. Firstly, attack patterns can specify only a sequence of events, rather than more complex forms. Secondly, there are no general purpose methods to prune the search except through the assertion primitives described above. And finally, they cannot detect denial of service attacks, failed logins, variations from normal usage, and passive listening — this is because these items are either not recorded by the audit trail mechanism, or they cannot be represented by state transition diagrams.

A small point to be noted is that USTAT was never meant to be a stand-alone

intrusion detection system; indeed, it is meant to be used with an anomaly detector so that more intrusion attempts may be detected by their combination. Some of the weaknesses of state transition systems are remedied by the Pattern Matching Model, discussed next.

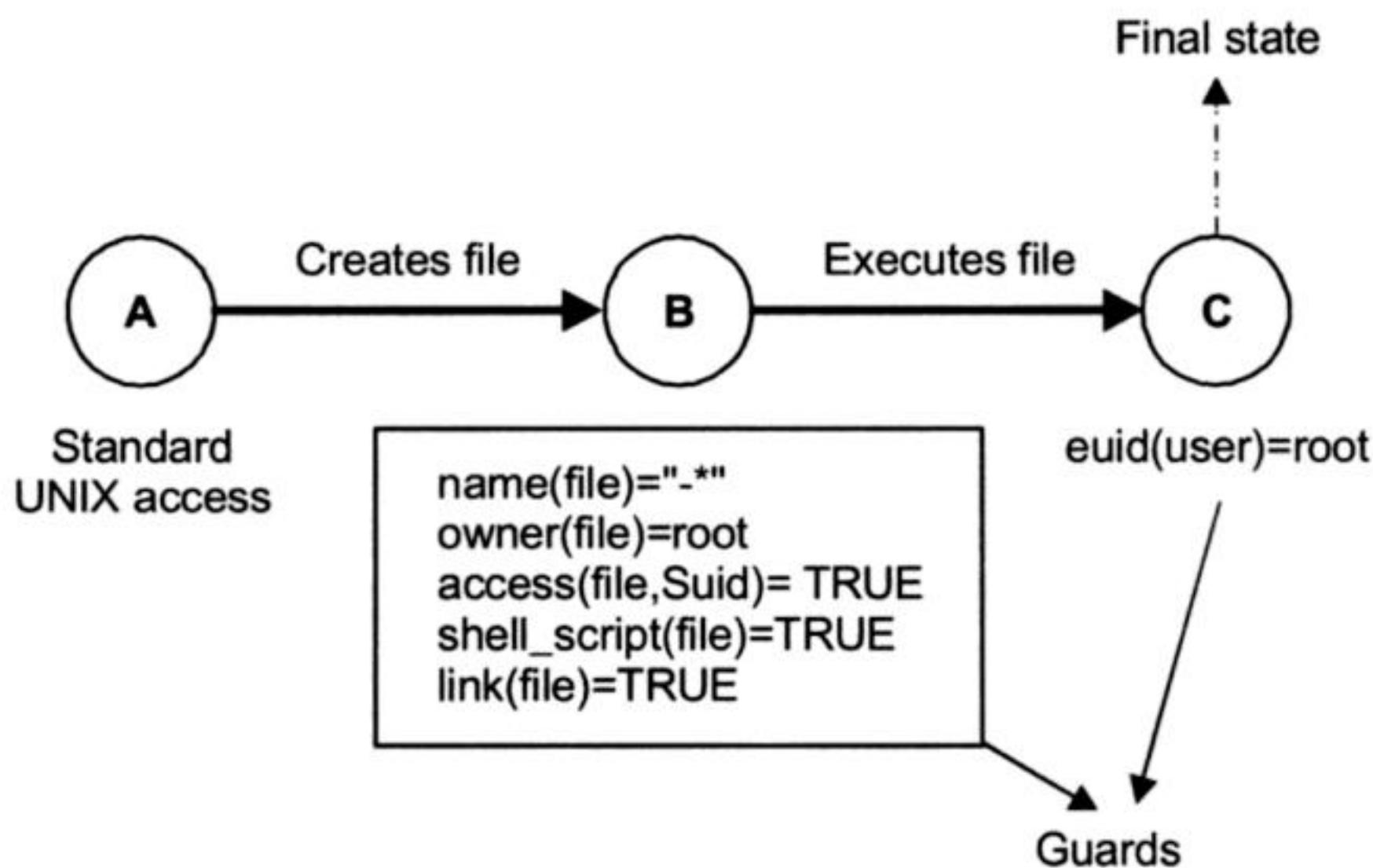


Figure 14.3 State transition diagram.

Kumar proposed a new misuse detection system based on Pattern Matching. This model encodes known intrusion signatures as patterns that are then matched against the audit data. Like the state transition analysis model, this model attempts to match incoming events to the patterns representing intrusion scenarios. The implementation makes transitions on certain events, called *labels*, and Boolean variables, called *guards*, can be placed at each transition. The difference between this and the state transition model is that the state transition model associates these guards with states, rather than transitions. The important advantages of this model are:

1. *Declarative specification.* It only needs to be specified what patterns need to be matched, not how to match them.
2. Multiple event streams can be used together to match against patterns for each stream without the need to combine streams. This means that streams can be processed independently, and their results can be analyzed together to give evidence of intrusive activity.
3. *Portability.* Since intrusion signatures are written in a system independent script, they need not be rewritten for different audit trails. The patterns' declarative specifications enable them to be exchanged across different operating systems and different audit trails.
4. It has excellent real-time capabilities. Kumar reports a CPU overhead of 5–6% when scanning for 100 different patterns, which is excellent.
5. It can detect some attack signatures like the failed logins signature that the state transition model cannot do.

One problem with this model is it can only detect attacks based on known vulnerabilities (a problem with misuse detection systems in general). In addition, pattern matching is not very useful for representing ill-defined patterns and it is not an easy task to translate known attack scenarios into patterns that can be used by the

The use of RBID systems requires the following:

- Personnel knowledgeable in rule-based systems, especially with respect to rule representation.
- Personnel who know how various activities may be represented in audit trails.
- Personnel experienced in intrusion detection and who have in-depth knowledge of the audit collection mechanism.

In addition to the costs associated with maintaining intrusion detection knowledge bases, there are several risks and limitations associated with this technology:

- Only known vulnerabilities and attacks are codified in the knowledge base. The knowledge base of rules is thus always playing “catch-up” with the intruders.
- The representation of intrusion scenarios—especially with respect to state-based approaches—is not intuitive.

For these reasons, RBIDs cannot detect all intrusion attempts.

Like all intrusion detection systems, RBIDs will negatively affect system performance due to their collecting and processing of audit trail information. For example, early prototyping of a real-time RBID system on a UNIX workstation showed the algorithm was using up to 50% of the available processor throughput to process and analyze the audit trail. Expert systems are an enabler for this technology.

14.7 DISTRIBUTED INTRUSION DETECTION

A distributed IDS (dIDS) consists of multiple Intrusion Detection Systems (IDS) over a large network, all of which communicate with each other, or with a central server that facilitates advanced network monitoring, incident analysis, and instant attack data. By having these cooperative agents distributed across a network, incident analysts, network operations and security personnel are able to get a broader view of what is occurring on their network as a whole. It combined the abilities of the Network Security Monitor (NSM) with intrusion detection monitoring of individual hosts.

A dIDS also allows a company to efficiently manage its incident analysis resources by centralizing its attack records and by giving the analyst a quick and easy way to spot new trends and patterns, and to identify threats to the network across multiple network segments. dIDS used a centralized analysis engine and required that agents be placed on the systems being monitored as well as in a place to monitor the network traffic.

14.7.1 Overview

The Central Analysis Server

The heart of the operation is the central analysis server which would ideally consist of a database and Web server. This allows the interactive querying of attack data for analysis as well as a useful Web interface to allow the corporate guys upstairs to see the current attack status of your network. It also allows analysts to perform pre-

programmed queries, such as attack aggregation, statistics gathering, to identify attack patterns and to perform rudimentary incident analysis, all from a Web interface.

The Cooperative Agent Network

The network agent is one of the most important components of the dIDS. An agent is the software process or device that does the actual data collection and inspection. It reports attack information to the central analysis server. The use of multiple agents across a network allows the incident analysis team a broader view of the network than can be achieved with single IDS systems.

Ideally, these agents will be located on separate network segments, and geographical locations (see Figure 14.4 below.) The agents can also be distributed across multiple physical locations, allowing for a single incident analysis team to view attack data across multiple corporate locations.

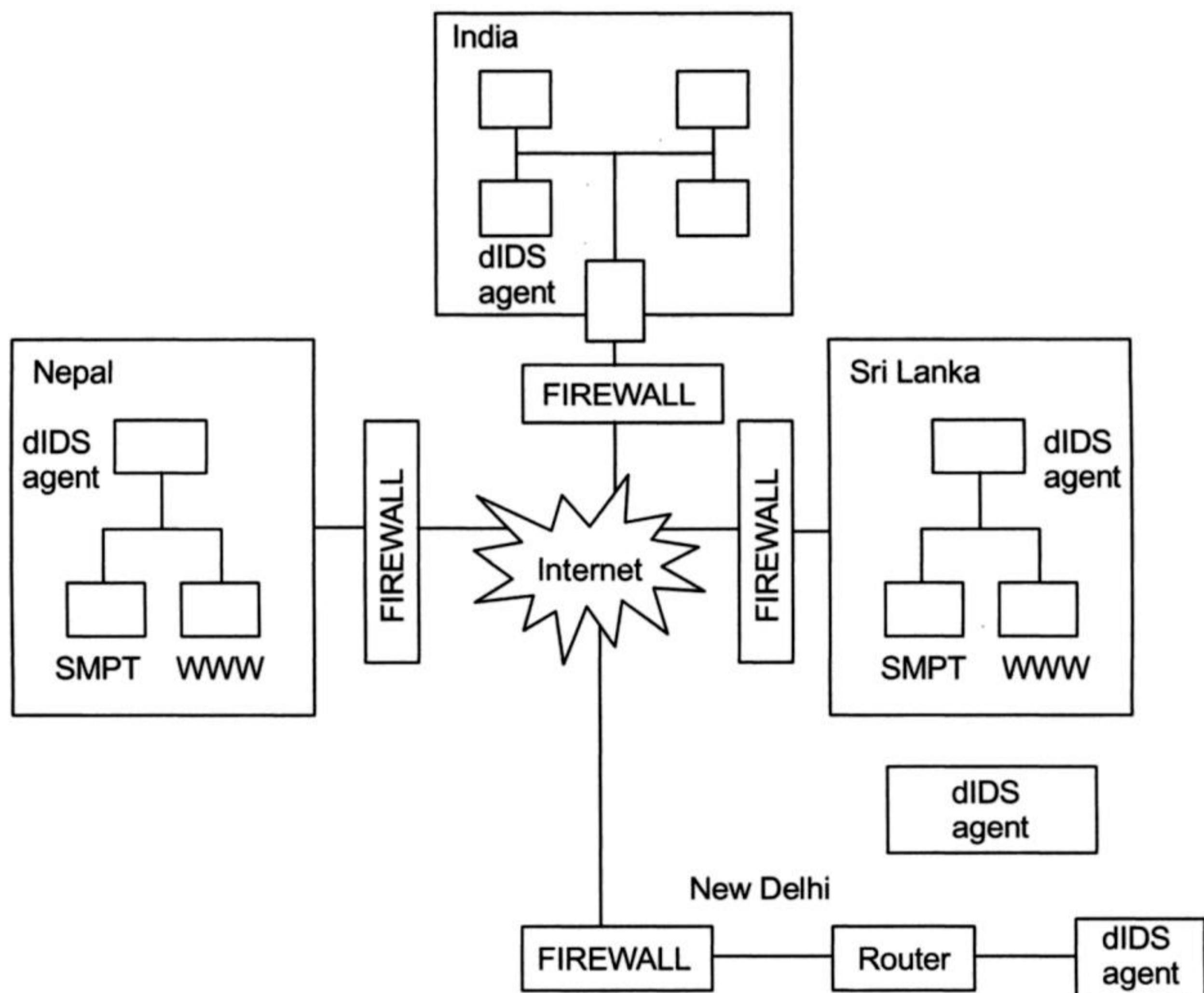


Figure 14.4 Agent network.

Although any IDS could be used on the agent machines, it is highly suggested that Snort be used. It has been demonstrated, however, that any attack logging system can be incorporated into this agent network. This can range from router attack logs to ipfw, firewalls, and even Windows personal firewall systems.

Attack Aggregation

Attack aggregation is another core part of the dIDS system. This part of the system is programming logic based on the central server. Aggregation simply refers to the method in which users group or order the information gathered from the agent network. One example of this would be to aggregate information according to attacker IP, putting all attacks from an attacking IP together with other attacks from the same IP. Another example is the aggregation of attack data according to destination (attacked) port, or even by date and time. Uses for aggregation will be explained later in this chapter.

14.7.2 Advantages of a dIDS

The dIDS offers the incident analyst many advantages over other single mode IDS systems. One of these advantages is the ability to detect attack patterns across an entire corporate network, with geographic locations separating segments by time zones or even continents. This could allow for the early detection of a well-planned and coordinated attack against the organization in question, which would allow the security people to ensure that targeted systems are secured and offending IPs are disallowed any access. Another proven advantage is to allow early detection of an Internet worm making its way through a corporate network. This information could then be used to identify and clean systems that have been infected by the worm, and prevent further spread of the worm into the network, thereby lowering any financial losses that would otherwise have been incurred.

The second major advantage is that a single analysis team can now do what previously required several incident analysis teams due to physical distance. This obviates the need to pay for distinct incident analysis teams for each separate geographic location of the organization's offices. Another issue that it addresses is attacks from within the corporations' network by angry, upset, or bored employees. By tying the central analysis server in with the companies DHCP or RADIUS servers, the incident analysts can track down people launching attacks from within the company, and track what they have attempted to do, as well as provide evidence against the perpetrators.

14.7.3 Incident Analysis with dIDS

Incident analysis using the dIDS system is really what it is all about. This is where all the power, potential, flexibility, and strength of the system as a whole lies. It is the reason why the dIDS was first conceptualized, to allow for advanced analysis of attacks occurring over multiple network segments, and at an advanced level.

14.7.4 Analysis Using Aggregation

Aggregation is the main component used to facilitate this advanced method of analysis across a network's multiple segments. By aggregating similar or related data, the analyst is able to easily see how an attack progressed through the different stages: from active network reconnaissance, to the final attack. It is possible for the incident analyst

to see what kind of time frame the attacker was working within and to correlate other attack attempts against the networks to determine if there were multiple cooperative attackers. The most common methods of aggregation are according to attacker IP, destination port, agent ID, date, time, protocol, or attack type.

- Aggregating by attacker IP allows the analyst to view the steps of an attacker's attempt from start to finish across the multiple network segments.
- Aggregating by destination port allows an analyst to view new trends in attack types, and to be able to identify new attack methods, or exploits being used.
- Aggregating by agent ID allows an analyst to see what variety of attacks and attackers have made attempts on the specific network segment the agent is on. Consequently, the analyst can determine if there are multiple attackers working in conjunction, or if there are network segments that are of more interest to attackers than others, thereby giving the security team a list of common targets to work on.
- Aggregating by date and time allows the analyst to view new attack patterns, and to potentially identify new worms or viruses that are only triggered at certain times.
- Aggregating by protocol helps in a purely statistical manner, which could allow an analyst to identify new attacks in particular protocols, or identify protocols on a network segment that should, under no circumstances, be there anyhow.
- Aggregating by attack type also allows for attack pattern matching and to correlate coordinated attacks against multiple network segments.

By utilizing all of these aggregation methods, the analyst is given an unlimited number of different sets of data to correlate against other attacks, detect coordinated distributed attacks, attacks from within their own network, and to detect new exploits and vulnerabilities being deployed by the underground hacking community.

The broad view given by the dIDS system also allows the analyst to ensure a minimum of false positives and false negatives by being able to see beyond a single network segment, into the network as a whole. For example, if the analyst saw that one out of five network segments got seven unrequested ICMP Echo packets, it could be a simple issue of false addressing or improper routing somewhere. However, if the analyst were to see that three separate network segments were reporting seven unrequested ICMP Echo packets, it is much more likely that these packets would be malicious in nature. This would cause the analyst to take note of the activity and perhaps check into the incident further or flag it for review at a later date.

14.8 BASE-RATE FALLACY

In order to apply base-rate fallacy reasoning in computer intrusion detection, we must first find the different probabilities, or if such probabilities cannot be found, make a set of reasonable assumptions regarding them.

14.8.1 Basic Frequency Assumptions

Let us, for the sake of further argument, hypothesize a figurative computer installation with a few tens of workstations, a few servers (all running UNIX), and a couple of dozen users. Such an installation could produce on the order of 1,000,000 audit records per day with some form of “C2” compliant logging in effect, in itself a testimony to the need for automated intrusion detection. Suppose further that, in such a small installation, we would not experience more than a few, say one or two, actual attempted intrusions per day. Even though it is difficult to get any figures for real incidences of attempted computer security intrusions, this does not seem to be an unreasonable number. Furthermore, assume that, at this installation, we do not have the man power to have more than one site security officer (SSO for short), who probably has other duties, and that the SSO, being only human, can only react to a relatively low number of alarms, especially if the false alarm rate is high (50% or so). Even though an intrusion could possibly affect only one audit record, it is likely, on an average, that it will affect a few more than that. Furthermore, a clustering factor actually makes our estimates more conservative, so it was deemed prudent to include one. Using data from a previous study of the trails that SunOS intrusions leave in the system logs, we can estimate that 10 audit records would be affected in the average intrusion.

14.8.2 Honeypots

A honeypot is a computer system on the Internet that is expressly set up to attract and “trap” people who attempt to penetrate other people’s computer systems. It is a class of powerful security tools that go beyond routine intrusion detection. Honeypots are closely monitored network decoys serving several purposes: they can distract adversaries from more valuable machines on a network, they can provide early warning about new attack and exploitation trends and they allow in-depth examination of adversaries during and after exploitation of a honeypot. When a collection of honeypots connects several honeypot systems on a subnet, it may be called a *honeynet*.

Honeypots are a highly flexible security tool with different applications for security. They don’t fix a single problem. Instead they have multiple uses, such as prevention, detection, or information gathering. Honeypots all share the same concept: a security resource that should not have any production or authorized activity. In other words, deployment of honeypots in a network should not affect critical network services and applications. A honeypot is a security resource whose value lies in being probed, attacked, or compromised.

The objectives of honeypot designs are:

- Divert an attacker from accessing critical systems
- Collect information about the attacker’s activity
- Encourage the attacker to stay on the system long enough for administration.

There are two general types of honeypots:

- *Production honeypots*: These are easy to use, capture only limited information, and are used primarily by companies or corporations.

Clearly, sound password management practices must take into consideration human limitations, to limit the above problems.

Composition Rules

As outlined in threats, one of the primary weaknesses of passwords is that they may be guessed. While a human may give up after trying to guess ten or a hundred possible passwords, software such as Crack and L0phtCrack will happily try millions of combinations.

To combat password guessing attack, users should pick hard-to-guess passwords. One way to do this is to ensure that the set of all possible passwords is too large to search thoroughly, and then to eliminate probably guesses.

The number of possible password combinations is calculated by taking the number of legal characters in a password, and raising that number to the number of characters in the password. The possibilities for some likely combinations are shown below.

Legal characters	5	6	7	8	9	10
0–9	1.00e05	1.00e06	1.00e07	1.00e08	1.00e09	1.00e10
a–z	1.00e06	3.09e08	8.03e09	2.09e11	5.43e12	1.41e14
a–z, 0–9	6.05e07	2.18e09	7.84e10	2.82e12	1.02e14	3.66e15
a–z, 0–9, 3 punct	9.02e07	3.52e09	1.37e11	5.35e12	2.09e14	8.14e15
a–z, A–Z	3.80e08	1.98e10	1.03e12	5.35e13	2.78e15	1.45e17
a–z, A–Z, 0–9	9.16e08	5.68e10	3.52e12	2.18e14	1.35e16	8.39e17
a–z, A–Z, 0–9, 32 punct	7.34e09	6.90e11	6.48e13	6.10e15	5.73e17	5.39e19

Users must be required to choose their passwords from the widest possible set of characters, subject to the constraints of the systems where those passwords reside. For example, most mainframes do not distinguish between uppercase and lowercase, and only allow three punctuation marks (fourth row in the table above).

You must then set a policy based on the smallest permissible set of legal password values — for example: 10 billion. To draw from the above table, mainframe compatible passwords must be at least seven characters long to meet the requirement of at least 10 billion possible passwords.

A reasonable set of password rules, designed to ensure that the search space for all possible values is as reasonably large, and that passwords are not too easy to guess, is as follows.

To ensure that the search space is sufficiently large: Passwords must be at least seven characters long. Passwords must contain at least one letter, and at least one digit. If this is compatible with your systems: Passwords must contain both uppercase and lowercase letters, and at least one punctuation mark or other ‘special’ character. To eliminate easily guessed passwords: Passwords must not be based on the user’s name or login ID.

Passwords must not be based on a dictionary word, in any language. Passwords may not contain more than two paired letters (e.g. abbcdde is valid, but abbbcd is not).

Changing and Reusing Passwords

Over time, passwords may be compromised in many ways, including:

Users may share passwords with friends or coworkers. Users may write them down, and they may then be exposed. Passwords may be guessed, either by humans or security diagnostic software. The servers that house passwords may be compromised, and their passwords acquired by an intruder. The networks that passwords travel between a user's workstation and servers that the user logs into may be compromised, and passwords may be recorded by an intruder during transmission.

Users may be tricked into providing their passwords to intruders via a social engineering effort. The help desk may be tricked into giving an intruder a valid password. To limit the usefulness of passwords that have been compromised, a best practice is to change them regularly. Common rules on many systems are to force users to change their passwords when they log in, if they have not been changed for an extended period (e.g. 30 or 60 days). In general, users should be required to change their passwords regularly, at most every 90 days, and preferably more frequently.

For the same reasons, users should not reuse old passwords, as they may already have been compromised. Many systems support this by recording some representation of old passwords, and ensuring that users cannot change their password back to a previously used value. When password history is enforced, users may figure out the number of passwords in their history file. As this number is normally 10 or fewer, a user who does not really wish to change his password when prompted to do so may make several consecutive password changes, and finally return his password to its original value.

To defeat such 'smart' users, some systems also enforce a password rule that limits the number of password changes that a user may make in any given day. By forcing users to spend several days, users are less inclined to defeat the password history mechanism. A better approach, though not yet available on many systems, is to simply maintain an unlimited number of entries in each user's password history. Since disk storage has become very cheap, this approach is now feasible.

Secrecy

Passwords are intended to uniquely identify a user. As such, they must be secrets—known only to the user they identify.

Users frequently behave in ways that lead to password disclosure. In particular, users may:

- Choose passwords which are easily guessed—so are not really secret.
- Share their passwords with coworkers, friends or family.
- Write down their passwords, and place the written password near their computer, or in an ostensibly private place like a wallet.

Corporate password management policy must forbid these practices, and allow for some negative consequence to users who violate these rules. In addition, alternative mechanisms should be provided, to help users manage passwords without writing them down or sharing. Key among these mechanisms is password synchronization, which helps users to remember rather than write down passwords.

Chapter 15

Malicious Software

15.1 MALICIOUS CODE

Malicious programs, often referred to as “Malware”, include computer viruses, worms, trojans, spyware, and other programs written specifically to spy on network traffic, record private communications, execute unauthorized commands, steal and distribute private and confidential information, disable computers, erase files, etc.

The various types of malicious programs are as follows:

1. Viruses
2. Worms
3. Trojans
4. Spyware

Each of these malicious programs is discussed in the following sections.

15.2 VIRUSES

The term’s ‘computer virus’ is often used to refer to any software that is intended to damage computer systems or networks. More narrowly, it is a piece of software designed to infect a computer system.

The threat of virus infections has increased dramatically in the past three to four years. Before the advent of e-mail attached viruses, viruses were spread through exchange of infected media and this limited the potential damage that a virus could result in. When e-mail-borne viruses appeared on the scene, the threat increased considerably. The sophistication of virus code has also contributed to the problem, as has the popularity of the Internet. Now viruses spread much faster and can potentially cause more damage than in the past. In the past, a virus infection could result in loss

on data on the infected computer and the inconvenience created by corrupted software on the infected machines. New forms of virus code have added the threat of loss of confidential information and individual privacy. Computer users are well advised to protect their computers from the threat of virus infections. Many organizations now require that their users use virus detection programs.

The possibilities of what viruses can do are almost limitless, but viruses can erase data on your computer; encrypt files; delete directory structures; prohibit you from using your computer; send files stored on your computer to contacts in your address book without your knowledge; and much more.

15.2.1 Types of Viruses

Viruses can be classified into several types as discussed below:

1. *Parasitic viruses.* A parasitic virus attaches itself to a file in order to propagate. It generally keeps most of the file intact and either adds itself to the start (prepending viruses) or end of the file (appending viruses). COM and EXE files are easiest to infect, as they are simply loaded directly into memory and execution always starts at the first instruction.
2. *Boot sector virus.* A virus that spreads when computers attempt to boot from infected floppy disks or when infected computers access floppy disks. Examples: Form, Disk Killer, Michelangelo, and Stone virus.
3. *Polymorphic virus.* It is a virus that changes its characteristics with each infection, making its detection more difficult. Examples: Involuntary, Stimulate, Cascade, Phoenix, Evil, Proud, Virus 101.
4. *Memory resident virus.* This is a virus which installs code in memory, which infects future programs.
5. *Stealth virus.* A stealth virus is a virus that hides its tracks after infecting the computer. Once the computer has been infected, the virus can make modifications to allow the computer to appear that it has not lost any memory and/or that the file size has not changed. Examples: Frodo, Joshi, Whale.
6. *Macro viruses.* A macro virus is a new type of computer virus that infects the macros within a document or template. When you open a word processing or spreadsheet document, the macro virus is activated and it infects the normal template (Normal.dot)—a general purpose file that stores default document formatting settings. Every document you open refers to the normal template, and hence gets infected with the macro virus. Since this virus attaches itself to documents, the infection can spread if such documents are opened on other computers. Examples: DMV, Nuclear, Word Concept.

Some viruses do little harm; others delete files, change the order or digits in entries in a spreadsheet, or disable the computer's operating system. Viruses are often designed to remain undetected until they have infected other programs or other computers. A time bomb describes a virus that is activated on a certain date or after a certain period of time. A logic bomb is activated by a certain sequence of events, such as the virus having replicated a specified

Prevention

First install virus protection software that can detect and eradicate viruses. Such programs are cheap and are easily available. The popular maxim that ‘prevention is better than cure’ is effectively applicable in this regard. Moreover, installed anti-virus software needs to be upgraded regularly to remain effective. Regular signature file upgradation is as important as installing an anti-virus software in itself, as new viruses emerge daily.

Businesses must take extra care to ensure that all of their Internet browsers are kept up to date. If, however, the browsers cannot be updated, users should sacrifice certain scripting features of JavaScript, Java and activeX. It is advisable to delete risky e-mail attachments without opening them and also to refrain from downloading files from the Internet.

Detection

To detect virus’s files on the computer or network should be scanned to find whether they bear similarities with known computer virus definitions. Anti-virus programs should be run regularly.

Eradication

Using real time monitoring anti-virus, the moment a virus is detected, warning is displayed on the screen. The virus must be countered with the help of virus protection programs that either repair the infected program or delete them. These simple rules can construct a bulwark against virus infection to your computer. Virus protection software can continue to do the job for us, but the onus lies on us to ensure that they have the required in built ability through timely upgradation.

15.2.3 Methods to Avoid Detection

In order to avoid detection by users, some viruses employ different kinds of deception. Some old viruses, especially on the MS-DOS platform, make sure that the “last modified” date of a host file stays the same when the file is infected by the virus. However, this approach does not fool anti-virus software.

Some viruses can infect files without increasing their sizes or damaging the files. They accomplish this by overwriting unused areas of executable files. These are called *cavity viruses*. For example, the CIH virus, or Chernobyl virus, infects portable executable files. Because those files had many empty gaps, the virus, which was 1 KB in length, did not add to the size of the file. Some viruses try to avoid detection by killing the tasks associated with anti-virus software before it can detect them.

As computers and operating systems grow larger and more complex, old hiding techniques need to be updated or replaced.

Avoiding Bait Files and other Undesirable Hosts

A virus needs to infect hosts in order to spread further. In some cases, it might be a bad idea to infect a host program. For example, many anti-virus programs perform an integrity check of their own code. Infecting such programs will therefore increase the

likelihood that the virus is detected. For this reason, some viruses are programmed not to infect programs that are known to be part of anti-virus software. Another type of hosts that viruses sometimes avoid is bait files. Bait files (or goat files) are files that are specially created by anti-virus software, or by anti-virus professionals themselves, to be infected by a virus. These files can be created for various reasons, all of which are related to the detection of the virus.

Anti-virus professionals can use bait files to take a sample of a virus (i.e. a copy of a program file that is infected by the virus). It is more practical to store and exchange a small, infected bait file, than to exchange a large application program that has been infected by the virus.

Anti-virus professionals can use bait files to study the behaviour of a virus and evaluate detection methods. This is especially useful when the virus is polymorphic. In this case, the virus can be made to infect a large number of bait files. The infected files can be used to test whether a virus scanner detects all versions of the virus. Some anti-virus software employs bait files that are accessed regularly. When these files are modified, the anti-virus software warns the user that a virus is probably active on the system.

Since bait files are used to detect the virus, or to make detection possible, a virus can benefit from not infecting them. Viruses typically do this by avoiding suspicious programs, such as small program files or programs that contain certain patterns of ‘garbage instructions’.

A related strategy to make baiting difficult is sparse infection. Sometimes, sparse infectors do not infect a host file that would be a suitable candidate for infection in other circumstances. For example, a virus can decide on a random basis whether to infect a file or not, or a virus can only infect host files on particular days of the week.

Stealth

Some viruses try to trick anti-virus software by intercepting its requests to the operating system. A virus can hide itself by intercepting the anti-virus software’s request to read the file and passing the request to the virus, instead of the OS. The virus can then return an uninfected version of the file to the anti-virus software, so that it seems that the file is “clean”. Modern anti-virus software employs various techniques to counter stealth mechanisms of viruses. The only completely reliable method to avoid stealth is to boot from a medium that is known to be clean.

Self-modification

Most modern anti-virus programs try to find virus patterns inside ordinary programs by scanning them for so-called virus signatures. A signature is a characteristic byte-pattern that is part of a certain virus or family of viruses. If a virus scanner finds such a pattern in a file, it notifies the user that the file is infected. The user can then delete, or (in some cases) “clean” or “heal” the infected file. Some viruses employ techniques that make detection by means of signatures difficult or impossible. These viruses modify their code on each infection. That is, each infected file contains a different variant of the virus.

and usually performs malicious actions, such as using up the computer's resources and possibly shutting the system down.

Worms use parts of an operating system that are automatic and usually invisible to the user. It is common for worms to be noticed only when their uncontrolled replication consumes system resources, slowing or halting other tasks. A computer worm differs from a computer virus in that a computer worm can run itself. A virus needs a host program to run, and the virus code runs as part of the host program. A computer worm can spread without a host program, although some modern computer worms also use files to hide inside.

15.3.1 Historical Background

The word 'worm' was carried from *The Shockwave Rider*, a science fiction novel published in 1975 by John Brunner. Researchers John F. Shoch and John A. Hupp of Xerox PARC chose the name in a paper published in 1982; (*The Worm Programs*, Comm ACM, 25(3):172–180, 1982), and it has since been widely adopted.

The first implementation of a worm was by these same two researchers at Xerox PARC in 1978. Shoch and Hupp originally designed the worm to find idle processors on the network and assign them tasks, sharing the processing load, and so improving the 'CPU cycle use efficiency' across an entire network. They were self-limited so that they would spread no farther than intended.

Though it was technically a Trojan horse, the Christmas Tree Worm was likely the first worm on a worldwide network, spreading across both IBM's own international network and BITNET in December 1987, bringing both networks to their knees. An early worm on the Internet, and the first to attract wide attention, was the Morris worm. It was also termed *The Internet Worm* by Peter Denning in an article in *American Scientist* (March–April, 1988) in which he distinguished between a virus and a worm, thereby becoming an early computer zoologist. His definition was more restricted than that of some other computer zoologists of the time. The Morris worm was written by Robert Tappan Morris, at the time a computer science graduate student at Cornell University, and released on November 2, 1988 using a friend's account on a Harvard University computer. It quickly infected large numbers of computers attached to the Internet and caused massive disruption. That it did not spread even farther and cause more trouble is largely due to some errors in its implementation. It propagated via several bugs in BSD Unix and related systems, and its component programs (including several versions of 'sendmail').

15.3.2 Different Types of Computer Worms

E-mail Worms

Spreading goes via infected e-mail messages. Any form of attachment or link in an e-mail may contain a link to an infected website. In the first case activation starts when the user clicks on the attachment, while in the second case activation starts when clicking the link in the e-mail.

documents via e-mail. A very common payload for worms is to install a backdoor in the infected computer to allow the creation of a “zombie” under control of the worm author—Sobig and Mydoom are examples which created zombies. Networks of such machines are often referred to as *botnets* and are very commonly used by spam senders for sending junk e-mail or to cloak their website’s address. Spammers are, therefore, thought to be a source of funding for the creation of such worms, and worm writers have been caught selling lists of IP addresses of infected machines. Others try to blackmail companies with threatened DoS attacks.

Backdoors, maybe installed, can be exploited by other malware, including worms. Examples include Doomjuice, which spreads using the backdoor opened by Mydoom, and at least one instance of malware taking advantage of the rootkit backdoor installed by the Sony/BMG DRM software they put on millions of music CDs ending in late 2005.

Worms with Good Intent

Whether worms can be useful is a common conundrum amongst theorists in computer science and artificial intelligence, beginning with the very first research into them at Xerox PARC. The Nachi family of worms, for example, tried to download, then install patches from Microsoft’s website to fix various vulnerabilities in the host system—the same vulnerabilities the Nachi worm itself exploited. This eventually made the systems affected more secure, but generated considerable network traffic (sometimes more than would have worms they were protecting against), rebooted the machine in the course of patching it, and, maybe most importantly, did its work without the explicit consent of the computer’s owner or user. As such, most security experts regard worms as malware, whatever their payload or their writers’ intentions.

15.3.3 Protecting against Computer Worms

Worms mainly spread by exploiting vulnerabilities in operating systems, or by tricking users to assist them. If all vendors supply regular security updates then the majority of worms are unable to spread to them. If a vendor acknowledges vulnerability but has yet to release a security update to patch it, a zero day exploit is possible, but these are relatively rare.

Users need to be wary of opening unexpected e-mail, and certainly should not run attached files or programs, or visit websites which such e-mails link to. However, as the ILOVEYOU showed long ago, and phishing attacks continue to prove, tricking a percentage of users will always be possible. Anti-virus and anti-spyware software are helpful, but must be kept up to date with new pattern files every few days at least.

15.4 TROJANS

15.4.1 Trojan Horses

Trojan horse programs are named after the use of a hollow wooden horse filled with enemy soldiers used to gain entry into the city of Troy in ancient Greece. A Trojan horse program is a useful or apparently useful program or command procedure

Figure 15.1 shows the block diagram of digital immune system.

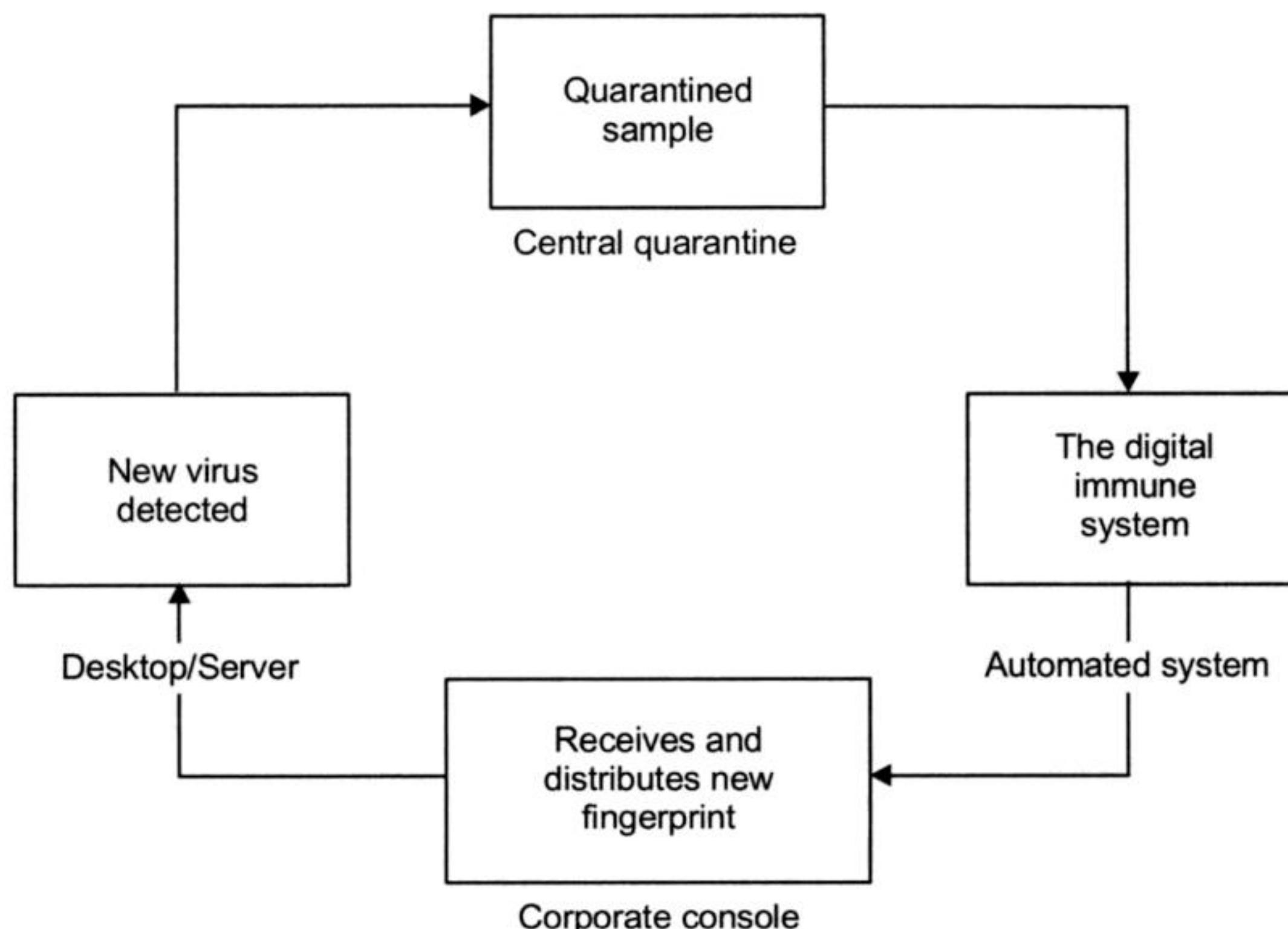


Figure 15.1 Digital immune system

15.8 ATTACKS

The objective of an attack on a cryptographic system is either to decipher the messages or to disrupt the network. There are many attacks on the computer which are the main threat for the security of the computer. Some of them are discussed here.

15.8.1 Hoax

A hoax is an attempt to trick an audience into believing that something false is real. There is often some material object (e.g., snake oil) involved which is actually a forgery; however, it is possible to perpetrate a hoax by making only true statements using unfamiliar wording or context. Unlike a fraud or con (which is usually aimed at a single victim and is made for illicit financial or material gain), a hoax is often perpetrated as a practical joke, to cause embarrassment, or to provoke social change by making people aware of something. Many hoaxes are motivated by a desire to educate by exposing the credulity of the public and the media or the absurdity of the target. Political hoaxes are sometimes motivated by the desire to ridicule or besmirch opposing politicians or political institutions, often before elections.

Characteristics of Hoaxes

- Hoaxes are not always created, initiated or sourced the same way
- Hoax by tradition
- Hoax by design (such as in war)
- Hoax originating in legitimate non-hoax use (e-mail hoax)
- Hoax by scare tactics (virus hoaxes)
- Urban legend

This is by no means a complete list; but the import is to show that hoaxes take many forms. The main characteristic of hoaxes is presenting the information or media as something real or believable to human understanding but is in fact false. Whether there is intent to deceive is not part of the hoax characteristics, as hoaxes are known both with and without it.

15.8.2 Back-door Attack

A back door is a means of access to a computer program that bypasses security mechanisms. A programmer may sometimes install a back door so that the program can be accessed for troubleshooting or other purposes. However, attackers often use back doors that they detect or install themselves, as part of an exploit. In some cases, a worm is designed to take advantage of a back door created by an earlier attack. For example, Nimda gained entrance through a back door left by Code Red. Whether installed as an administrative tool or a means of attack, a back door is a security risk, because there is always crackers out there looking for any vulnerability to exploit. Such an attack is usually used as either an access or modification attack. There are a number of tools available to create back door attacks on systems.

15.8.3 Brute Force Attack

A brute force attack consists of trying every possible code, combination, or password until you find the right one.

The difficulty of a brute force attack depends on several factors, such as:

- How long can the key be?
- How many possible values can each component of the key have?
- How long will it take to attempt each key?
- Is there a mechanism which will lock the attacker out after a number of failed attempts?

As an example, imagine a system which only allows 4 digit PIN codes. This means that there are a maximum of 10,000 possible PIN combinations.

How to Increase Security against a Brute Force Attack

PIN security could be increased by:

- Increasing the length of the PIN.
- Allowing the PIN to contain characters other than numbers, such as * or #
- Imposing a 30-second delay between failed authentication attempts.
- Locking the account after 5 failed authentication attempts.

A brute force attack will always succeed, eventually. However, brute force attacks against systems with sufficiently long key sizes may require billions of years to complete.

The attacker must monitor the packets sent from Archana to Samiksha and then guess the sequence number of the packets. Then the attacker knocks out Archana with a SYN attack and injects his own packets, claiming to have the address of Archana. Archana's firewall can defend against some spoof attacks when it has been configured with knowledge of all the IP addresses connected to each of its interfaces. It can then detect a spoofed packet if it arrives at an interface that is not known to be connected to the IP address. Many carelessly designed protocols are subject to spoof attacks, including many of those used on the Internet.

URL Spoofing and Phishing

Another kind of spoofing is “web page spoofing”, also known as *phishing*. In this attack, a legitimate web page such as a bank’s site is reproduced in “look and feel” on another server under control of the attacker. The intent is to fool the users into thinking that they are connected to a trusted site, for instance to harvest user names and passwords.

This attack is often performed with the aid of URL spoofing, which exploits web browser bugs in order to display incorrect URLs in the browsers location bar; or with DNS cache poisoning in order to direct the user away from the legitimate site and to the fake one. Once the user puts in his password, the attack-code reports a password error, then redirects the user back to the legitimate site.

Referer Spoofing

Some websites, especially pornographic paysites, allow access to their materials only from certain approved (login-) pages. This is enforced by checking the referer header of the HTTP request. This referer header, however, can be changed (known as “Referer spoofing” or “Ref-tar spoofing”), allowing users to gain unauthorized access to the materials.

Poisoning of File-sharing Networks

“Spoofing” can also refer to copyright holders placing distorted or unlistenable versions of works on file-sharing networks, to discourage downloading from these sources.

Caller ID Spoofing

In public telephone networks, it has for a long while been possible to find out who is calling you by looking at the Caller ID information that is transmitted with the call. There are technologies that transmit this information on landlines, on cellphones and also with VoIP. Unfortunately, there are now technologies (especially associated with VoIP) that allow callers to lie about their identity, and present false names and numbers, which could of course be used as a tool to defraud or harass. Because there are services and gateways that interconnect VoIP with other public phone networks, these false Caller IDs can be transmitted to any phone on the planet, which makes the whole Caller ID information now next to useless. Due to the distributed geographic nature of the Internet, VoIP calls can be generated in a different country to the receiver, which means that it is very difficult to have a legal framework to control those who would use fake Caller IDs as part of a scam.

15.8.6 Denial-of-service Attack (DoS Attack)

Denial-of-service attack is a type of attack on a network that is designed to bring the network to its knees by flooding it with useless traffic. Many DoS attacks, such as the Ping of Death and Teardrop attacks, exploit limitations in the TCP/IP protocols. For all known DoS attacks, there are software fixes that system administrators can install to limit the damage caused by the attacks. But, like viruses, new DoS attacks are constantly being dreamed up by hackers.

How a “Denial-of-service” Attacks Works?

In a typical connection, the user sends a message asking the server to authenticate it. The server returns the authentication approval to the user. The user acknowledges this approval and then is allowed onto the server.

In a denial-of-service attack, the user sends several authentication requests to the server, filling it up. All requests have false return addresses, so the server cannot find the user when it tries to send the authentication approval. The server waits, sometimes more than a minute, before closing the connection. When it does close the connection, the attacker sends a new batch of forged requests, and the process begins again—tying up the service indefinitely.

Blocking a “Denial-of-service” Attack

One of the more common methods of blocking a “denial-of-service” attack is to set up a filter, or “sniffer”, on a network before a stream of information reaches a site’s Web servers. The filter can look for attacks by noticing patterns or identifiers contained in the information. If a pattern comes in frequently, the filter can be instructed to block messages containing that pattern, protecting the Web servers from having their lines tied up.

15.8.7 Distributed Denial-of-service Attack

Distributed denial-of-service attack has recently emerged as one of the most newsworthy, if not the greatest, weaknesses of the Internet. On the Internet, a distributed denial-of-service (DDoS) attack is one in which a multitude of compromised systems attack a single target, thereby causing denial of service for users of the targeted system. The flood of incoming messages to the target system essentially forces it to shut down, thereby denying service to the system to legitimate users. DDoS attacks involve breaking into hundreds or thousands of machines all over the Internet. Then the attacker installs DDoS software on them, allowing them to control all these burgled machines to launch coordinated attacks on victim sites. These attacks typically exhaust bandwidth, router processing capacity, or network stack resources, breaking network connectivity to the victims. A hacker (or, if you prefer, cracker) begins a DDoS attack by exploiting a vulnerability in one computer system and making it the DDoS “master”. It is from the master system that the intruder identifies and communicates with other systems that can be compromised. The intruder loads cracking tools available on the Internet on multiple—sometimes thousands of—compromised systems. With a single command, the intruder instructs the controlled machines to launch one



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

is a perplexing problem. It highlights the impotence of standard access controls, because authorized users are requesting authorized actions. Effective computer security policies and practices can do much to eliminate the spread of viruses and worms. However, nothing can do more to stop the spread of pathogenic programs than educating and training users in virus prevention.

EXERCISES

1. What is a malicious program? List the various types of malicious programs.
2. List the various types of viruses. Explain each in detail.
3. Explain the working of e-mail viruses.
4. How does anti-virus software work?
5. What is worm? What are the different types of worms? Explain the working of worm.
6. What is Trojan horse?
7. Explain Spyware.
8. Explain digital immune system.
9. What are the different types of attack? Explain each in brief.

Chapter 16

Firewall

16.1 INTRODUCTION

A firewall is a system designed to prevent unauthorized access to or from a private network. It can be an effective means of protecting a local system or network of systems from unauthorized network users, at the same time affording access to the outside world via WAN and the Internet. Firewalls are one of the first lines of defence in a network. They can be implemented in both hardware and software, or a combination of both. Firewalls are frequently used to prevent unauthorized Internet users from accessing private networks connected to the Internet, especially intranets. All messages entering or leaving the intranet pass through the firewall, which examines each message and blocks those that do not meet the specified security criteria. The basic purpose of a firewall is to isolate one network from another.

Networks with internet access require security controls that include: encryption, one-time passwords and firewalls. Firewall principle and practice belongs to both access control and network security theory. The main function of a firewall is to centralize access control. Routers can filter on the IP address of data packets (up through the IP layer) while a firewall can stop all data packets and filter up through the application layer.

Firewalls are most often configured to be transparent to internal-network users and non-transparent to outside-network users. They are often installed between the network of an entire organization and the Internet, but could also be installed in an intranet to protect individual departments. Newer firewall products are not network addressable, that is, they require physical access via a terminal to change configuration.

Protections associated with firewall systems are as follows:

- Block unwanted traffic
- Direct incoming traffic to more trustworthy internal systems
- Hide vulnerable internal-network systems

- Hide internal network information, such as system names, network topology, network device types, internal user id's, etc.
- Provide more robust user authentication.

Most corporations are not ready for their first firewall, often encountering:

- Documentation is not adequate.
- Lost IDs on outside servers once the firewall assumes single IP address for the entire network.
- Certain network utilities will no longer work if the firewall is configured properly: PING, TRACEROUTE.

16.1.1 Characteristics of a Firewall

Following are the characteristics of a firewall:

- All traffic between two networks must pass through the firewall.
- Only traffic that is authorized by local security policy is permitted to pass.
- The firewall itself is immune to penetration.
- A firewall cannot guarantee protection from outside attacks.
- Firewall implementation requires risk analysis to define the level of protection.
- Firewall policy is a component of local security policy. It
 - defines the level of protection to be expected from the firewall,
 - specifies the process by which requests for exceptions will be considered, and
 - defines what is authorized traffic and/or what is denied traffic.

Firewalls operate on a single rule which may be either of the two given below:

- All traffic is denied except that which is specifically authorized.
- All traffic is allowed except that which is specifically denied.

There are three main classes of firewalls: packet filters, application level gateways and circuit (proxies) gateways.

16.2 PACKET FILTERS

Packet filters firewalls were the first generation of firewalls. It works at OSI layers 3 and 4. Figure 16.1 shows the packet filters. Packet filters track the source and destination address of IP packets, and source and destination of TCP/UDP port numbers, permitting packets to pass through the firewall based on rules that the network manager has set. The packet filter does not analyze the contents of a packet. A packet filter firewall can allow any traffic that you specify as acceptable.

There are various strategies for implementing packet filters. Some of them are:

- Build rules from most to least specific order
- The most specific rule should be placed near the top of the rule set.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

Let us take an example. In this matrix example, there exists two processes, a file and some device. The first process has the ability to execute the second, read the file and write some information to the device, while the second process can only send information to the first.

	Process 1	Process 2	Device	File
Process 1	Read, Write, Execute	Read	Read	Write
Process 2	Read, Execute	Read, Write, Execute		

The access control matrix can be used as a model of the static access permissions in any type of access control system. It does not model the rules by which permissions can change in any particular system, and therefore only gives an incomplete description of the system's access control security policy.

An access control matrix should be thought of only as an abstract model of permissions at a given point in time; a literal implementation of it as a two-dimensional array would have excessive memory requirements. Capability-based security and access control lists are categories of concrete access control mechanisms whose static permissions can be modelled using access control matrices. Although these two mechanisms have sometimes been presented as simply row-based and column-based implementations of the access control matrix, this view has been criticized as drawing a misleading equivalence between systems that do not take into account dynamic behaviour. Rules can be applied to people, as well as devices.

SUMMARY

Firewalls make it possible to filter incoming and outgoing traffic that flow through your system. They can greatly enhance the security of a host or a network. They can be used to protect and insulate the applications, services and machines of your internal network from unwanted traffic coming from the public Internet. They can be used to limit or disable access from hosts of the internal network to services of the public Internet and also used to support network address translation. Anyone who is responsible for a private network that is connected to a public network needs firewall protection. There are three main classes of firewalls: packet filters, application and circuit gateways (proxies), and stateful inspection (or smart filter) firewalls. Firewalls can either be software based or they can be a piece of physical hardware that acts as a gateway. A good firewall keeps personal data and hackers out.

Firewalls introduce problems of their own. They can also constitute a traffic bottleneck. They concentrate security in one spot, aggravating the single point of failure phenomenon. But the major benefit of the firewalls is that they protect private local area networks from hostile intrusion from the Internet.

Chapter 17

Computer Forensics

17.1 INTRODUCTION

Computer forensics, still a rather new discipline in computer security, focuses on finding the digital evidence after a computer security incident has occurred. It is the analysis of information contained within and created with computer systems and computing devices, typically in the interest of figuring out what happened, when it happened, how it happened, and who was involved. Computer forensics is the process of investigating a computer system to determine the cause of an incident. It relates to the application of scientific knowledge to legal problems and is useful mostly for crime detection system and for investigation of the various computer related crimes.

Computer forensics requires specialized expertise and knowledge that goes beyond normal data collection and preservation techniques available to the users. Forensics investigations require special training and skills. This process often involves investigating computer systems to determine whether they are or have been used for illegal or unauthorized activities. Mostly, computer forensics experts investigate data storage devices, like hard disks or CDs.

The goal of computer forensics is to do a structured investigation and find out exactly what happened on a digital system, and who was responsible for the same.

For recovering evidence from a computer system or storage medium, the following phases are used:

Identification of the sources of documentary or other evidence: In this phase, the investigator identifies the possible containers of computer related evidence, such as hard drives, CDs, floppy disks, and log files, to name a few. A computer or hard drive itself is not evidence—it is a possible container of evidence. We use these sources in the analysis phase, for identifying the information and data that is actually pertinent to the situation at hand.

Preservation: Preservation of evidence requires limited access. When performing a computer forensics analysis, we must do everything possible to preserve the original media and data. Typically, this involves making a forensic image or forensic copy of the original media, and conducting our analysis on the copy versus the original. It is a good idea to seal evidence in a bag and identify the date, time, and person responsible for collection of it.

Extraction: Any evidence found relevant to the situation at hand will need to be extracted from the working copy media and then typically saved to another form of media as well as printed out. Each time the evidence is handled, it should be preserved properly.

Interpretation: This is the most challenging and interesting part of the forensics process. Understand that just about anyone can perform a computer forensics “analysis”. Some of the GUI tools available make it extremely easy. Being able to find evidence is one thing, the ability to properly interpret it is another story. To accomplish an analysis, you must understand the operating system and application that you are investigating. You should make sure that you do not write data to the disk, because doing so may destroy the evidence. We will cite one example.

The experts for the prosecution in a case used a popular GUI tool that came with a script for finding Internet search engine activity. When they ran the script, they found literally hundreds and hundreds of “searches” that supposedly had been conducted by the defendant. Therefore, the defendant had intentionally accessed certain types of information related to these searches—the searches showed intent.

When the experts for the defence examined the same evidence, they realized that each and every one of these “searches” was actually a hyperlink and not a search at all. The hyperlinks were formed in such a way that when a link was clicked, a database was searched to pull up the most current information related to the link. The way that the links within the page were formed was what the GUI tool honed in on, as they were formed similarly to fragments and web pages that could be found to indicate search engine activity.

The experts for the prosecution took for granted that their automated tool was accounting for any variables, and would only show them searches that had actually been conducted a big mistake. These experts lacked the technical skills to authenticate their results, so they depended entirely on a single automated tool.

This leads to a very important lesson. Results from any tool should always be thoroughly checked by someone versed in the underlying technology to see if what appears to be a duck is actually a duck.

In the very same case, the experts for the defence recovered reams of e-mail that the prosecution experts did not find. This was due to the fact that the prosecution experts simply did not know how to find it.

It is interesting to note that both the experts for the defense and the prosecution used the same primary tool in their analysis. The differences in what was found by one side versus the other, as well as the differences in analysing was due to the experience and education levels of the experts—it had nothing to do with the tool being used.

Factual reporting of the information found: Your findings and reports need to be based on proven techniques and methodology, and you as well as any other competent forensic examiner should be able to duplicate and reproduce the results.

Providing expert opinion: You may have to testify or relate your findings and opinions about your findings in a court of law or other type of legal or administrative proceeding.

Computer forensics can be used to uncover potential evidence in many types of cases including, for example:

- Copyright infringement
- Industrial espionage
- Money laundering
- Piracy
- Sexual harassment
- Theft of intellectual property
- Unauthorized access to confidential information
- Blackmail
- Corruption
- Decryption
- Destruction of information
- Fraud
- Illegal duplication of software
- Unauthorized use of a computer
- Child pornography

17.2 COMPUTER FORENSICS INVESTIGATIONS

Computer forensics investigations can be divided into two primary types as given below:

1. When the computer(s) was/were used as an instrument to commit a crime or involved in some other type of misuse. In this, you may or may not be present when the computing device is shut down to begin an investigation. You may have hard drives and other media delivered to you to analyze.
2. When the computer is used as the target of a crime—hacked into and information stolen for example. When computer forensics techniques and methodology are used in this situation to figure out what happened, we typically call this incident response. In this type of investigation, you will typically always want to capture information that is extremely volatile, such as information contained in RAM concerning network connections and running processes.

Regardless of the situation, and whether the evidence will be used in a court of law or as the grounds for a letter of reprimand, the techniques, procedures, and methodologies used should be largely the same.

17.2.1 Types of Data

In computer forensics, there are three types of data that we are concerned with—

active, archival, and latent. Active data is the information that anybody can see, for example data files, programs, and files used by the operating system. This is the easiest type of data to obtain. Archival data is the data that has been backed up and stored. This could consist of backup tapes, CDs, floppies, or entire hard drives, to cite a few examples. Latent data is the information that one typically needs specialized tools to get at. An example would be information that has been deleted or partially overwritten.

A computer investigation could entail looking at one or more of these data types depending on the circumstances. Obtaining latent data is by far the most time-consuming and costly.

17.3 AREAS OF APPLICATION OF COMPUTER FORENSICS

Computer forensics is applied in both public and private sectors.

17.3.1 Public Sector

Computer forensics is used in the public sector by government and law enforcement personnel to investigate and prosecute crimes. Criminals are using computer technology when committing “traditional” crimes such as homicide, rape, fraud, and auto theft. They are also using computer technology to commit crimes that would not be possible without computing devices, such as breaking into a networked system and stealing or altering data, posting child pornography to a newsgroup, or harassing someone via e-mail.

Computers can be the target of a crime (your computer system is attacked over the Internet), the tool in the commission of a crime (sending and receiving child pornography), or as incidental to a crime (keeping records concerning the houses you have burgled). When computing devices are used in committing crimes, you will often hear the term “Cyber crime” used. Although the word “Cyber” does get people’s attention, it is often misused—Cyber typically denotes being online. You are not in “Cyberspace” just by turning your computer on.

At any rate, use of computer forensics by government and law enforcement is increasing, as more and more criminals are using computing technology. Computer evidence is used by prosecutors every day to aid in convicting criminals involved in fraud, murder, drug trafficking, child pornography, embezzlement, and terrorism.

17.3.2 Private Sector

In the private sector, computer forensic techniques and methodologies are used to investigate electronic break-ins, embezzlement, improper use of computing resources by employees, and theft of trade secrets among other things.

Those in the insurance business may use information retrieved from computer systems to identify fraud in workman’s compensation, automobile or personal accident cases, or arson. I am aware of a few cases where e-mails were sent outlining plans to fake back injuries and other ailments in order to receive money from the insurance company. These e-mails were used to convict those making the false claims.

Evidence gleaned from a forensic investigation and examination is not limited to what is found or extracted from magnetic media such as hard drives, floppy drives, and tapes. Evidence can be in the form of visual output on a computer monitor, printouts, and passwords written down, notes made in computer or software manuals, or logs from systems external to the subject computer itself, such as proxy servers or firewalls. The computer forensics practitioner that limits himself to looking at only the magnetic media on the subject computer will be missing important clues.

A computer forensics practitioner must always remember that there might be, and probably is, evidence related to the situation that is external to the computer itself. In some situations, this external evidence could not only make or break the case, it might even be the best evidence that you can obtain.

Computer forensics is done in a fashion that adheres to the standards of evidence that are admissible in a court of law. Thus, computer forensics must be techno-legal in nature rather than purely technical or purely legal.

17.4 UNDERSTANDING THE SUSPECTS

It is absolutely essential for the forensics team to have a perfect understanding of the level of sophistication of the suspect(s). If insufficient information is available to form this opinion, the suspects must be considered to be experts, and should be presumed to have installed countermeasures against forensic techniques. Because of this, it is critical that you appear to the equipment to be as indistinguishable as possible from its normal users until you have shut it down completely, either in a manner which probably prohibits the machine modifying the drives, or in exactly the same way they would.

If the computer contains only a small amount of critical data on the hard drive, for example, software exists to wipe it permanently and quickly if a given action occurs. It is straightforward to link this to the Microsoft Windows “Shutdown” command, for example. However, simply “pulling the plug” is not always a great idea, either the information stored solely in RAM, or on special peripherals, may be permanently lost. Losing an encryption key stored solely in RAM, and possibly unknown even to the suspects themselves by virtue of having been automatically generated, may render a great deal of data on the hard drive(s) unusable, or at least extremely expensive and time-consuming to recover.

17.4.1 Electronic Evidence Considerations

Electronic evidence can be collected from a variety of sources. Within a company’s network, evidence will be found in any form of technology that can be used to transmit or store data. Evidence should be collected through three parts of an offender’s network: at the workstation, on the server accessed, and on the network that connects the two. Investigators can therefore use three different sources to confirm the origin of the data.

Like any cases, the information generated as the result of a computer forensics investigation must follow the standards of admissible evidence. Special care must be taken when handling a suspect’s files; dangers to the evidence include viruses,

You should specifically look for a wire running from anything to the CMOS battery or “CMOS clear” jumper. CMOS memory can be used to store data on the motherboard itself, and if power is removed from it, the contents will be lost. You must avoid causing CMOS memory to lose power. Encryption keys, etc. may be stored here.

Once you have determined that the case is safe to open, proceed to remove the cover.

Fully Document Hardware Configuration

Completely photograph and diagram the entire configuration of the system. Note serial numbers and other markings. Pay special attention to the order in which the hard drives are wired, since this will indicate boot order, as well as being necessary to reconstruct a RAID array. A little time being thorough here will save you more later.

Duplicate the Hard Drives

Using a stand-alone hard-drive duplicator or similar device, completely duplicate the entire hard drive. This should be done at the sector level, making a bit-stream copy of every part of the user-accessible areas of the hard drive which can physically store data, rather than duplicating the file system. Be sure to note which physical drive each image corresponds to. The original drives should then be moved to secure storage to prevent tampering.

Use some kind of hardware write protection to ensure no writes will be made to the original drive. Even if operating systems like Linux can be configured to prevent this, a hardware write blocker is the best practice. The process is often called *imaging*. You can image to another hard disk drive, a tape, or other media. Tape is a preferred format for archive images, since it is less vulnerable for damage and can be stored for a longer time.

There are two goals when making an image:

- Completeness (imaging all of the information)
- Accuracy (copying it all correctly)

The imaging process is verified by using the SHA-1 message digest algorithm or other still viable algorithms. To make a forensically sound image, you need to make two reads that result in the same output by the MD algorithm. Generally, a drive should be hashed in at least two algorithms to help ensure its authenticity from modification in the event one of the algorithms is cracked. This can be accomplished by first imaging to one tape labelled as the *Master* and then make an image labelled *Working*. If onsite and time is critical, the second read can be made to *Null*.

Ultimately, the methodology used by computer forensic investigators in capturing potential evidence on a system will be dictated by the proportionality of the likely importance of that evidence in the matter for which these services are engaged. Additional influences such as claims of privilege and potential damages sought for business interruption create potential headaches for corporate investigations where forensic soundness is often sacrificed for practicality. Law enforcement personnel moving into the corporate environment tend to be overly strict in their application of computer forensic principles in litigations where the burden of proof does not require it. There is an increasing need to capture servers live and capturing less than whole

disks worth of data in an effort to work within a time and cost framework. Even an unsolved murder investigation must be wound up at some point where there are diminishing gains to be had in progressing the investigation, so too with computer forensic investigations in both the corporate and criminal arenas where the sheer quantity of digital evidence can become overwhelming and threaten to overburden investigators. Also, it must be remembered that any computer evidence is potentially admissible regardless of the methodology by which it came to the attention of the court. In other words, if you forget to get a SHA or MD5 hash on the original hard drive, do not for a minute think that the data is worthless or non admissible, traditional discovery has been happening for at least a decade without a second thought to hashes. Application of proper forensic principles will, however, improve its overall credibility and diminish admissibility challenges.

17.4.2 E-mail Review

E-mail has become one of the primary mediums of communication in the digital age, and vast amounts of evidence may be contained therein, whether in the body or enclosed in an attachment. Because users may access e-mail in a variety of ways, it is important to look for different kinds of e-mails. The user may have used a dedicated program, or Mail User Agent (MUA), a web browser, or some other program to read and write e-mail. Additionally, files for each of these programs may be stored on a local hard drive, a network device, or a removable device. A good examiner will search all of these locations for e-mail data. Be aware that many e-mail clients will save a copy of outgoing messages, so both the sender and the recipient may have a copy of each message. Finally, mail may also be stored on a dedicated mail server, either awaiting delivery or as permanent storage.

E-mail Headers

Main article: E-mail#Internet e-mail header

All e-mail programs generate headers that attach to the messages. The study of these headers is complex. Some investigators favour reading the headers from the bottom up, others from the top down. Under normal circumstances, headers are supposed to be created by the mail user agent and then prepended by mail servers, the bottom up method should work. But a malicious mail server or forger may make this difficult.

The headers added by an MUA are different from those added by mail servers. For example, here is the format for headers generated by Mozilla Thunderbird 1.0 running on Microsoft Windows:

```
Message-ID: <31B5F981.5040504@hostname.net>
Date: Tue, 11 Aug 2007 13:42:09 -0500
From: User Name <username@hostname.net>
User-Agent: Mozilla Thunderbird 1.0 (Windows/20041206)
X-Accept-Language: English
MIME-Version: 1.0
```

To: recipient@otherhost.com

Subject: Testing

Content-Type: text/plain; charset=ISO-8859-1; format=flowed

Content-Transfer-Encoding: 7bit

Extensions such as enigmail may add extra headers.

The Message-ID field has three parts:

The time the message was sent in seconds past the epoch in hexadecimal.

A random value called a *coep*. The coep is of the format #0#0#0# where # is a random digit. Because Thunderbird treats the coep like a number, it may be shorter if the leading digits are zeros. For example, a coep of "0030509" would display as "30509".

The fully qualified domain name of the sender is:

Message-ID: [time].[coep]@[domain-name]

Information on the Message-ID header was derived from the source code in mozilla/mailnews/compose/src/nsMsgCompUtils.cpp in function msg_generate_message_id() and therefore applies only to mail sent by this application. Generally, the format of the Message-ID is arbitrary, and you should refer to the applicable RFCs.

Sorting through the Masses

While theoretically possible to review all e-mails, the sheer volume that may be subject to review may be a daunting task; large-scale e-mail reviews cannot look at each and every e-mail due to the sheer impracticality and cost. Forensics experts use review tools to make copies of and search through e-mails and their attachments, looking for incriminating evidence using keyword searches. Some programs have been advanced to the point that they can recognize general threads in e-mails by looking at word groupings on either side of the search word in question.

17.5 EXAMPLES OF COMPUTER FORENSIC

A few examples of computer forensic are given below:

1. There have been a number of cases recently found at private schools where authority figures have been charged with possession of child pornography. These discoveries were made from the use of computer forensics. Through the ability to track the buying and selling of pornography online, computer forensic investigators have been able to locate people involved with these crimes. They are able to use this information they have found on the computers as circumstantial evidence in court, allowing prosecution to occur. Due to this profession, child pornographers are being penalized for their actions and taking them out of the education system.
2. Another example of how computer forensics is affecting the current workplace is the aspect of security. Employees' work computers are now being monitored to ensure no illegal actions take place in office. They also have heightened security, so outsiders cannot access a company's confidential files. If this security is broken, a company is then able to use computer forensics to trace

back to which computer was being tampered with and what information was extracted from it, possibly leading to the guilty parties and other potential parties involved.

17.6 FREE SPACE AND SLACK SPACE

Free space is the unused clusters on a hard disk. It is also the space between the end of a file and the end of the disk cluster it is stored in. Also called “file slack,” it occurs naturally because data rarely fill fixed storage locations exactly, and residual data occur when a smaller file is written into the same cluster as a previous larger file. In computer forensics, slack space is examined because it may contain meaningful data. When we delete a file on the hard disk, the file is not actually deleted. Instead, a pointer in a file allocation table is deleted. This pointer was used by an operating system to track down the file when it was referenced, and the act of deleting the file merely removes the pointer and marks the sector(s) holding the file as available for the operating system to use. The original data stored remains on the disk. Since the deleted file is not actually completely erased or overwritten, it just sits there until the operating system needs to use that space for another file or application. Sometimes, the second file that is saved in the same area does not occupy as many sectors as the first file, so there will still be a fragment of the original file. The sector that holds the fragment of this file is referred to as *free space* because the operating system has marked it usable when needed. When a file is saved to a storage media, such as a hard drive, the operating system allocates space in blocks of a predefined size, called *sectors*. The size of all sectors is the same on a given system or hard drive. Even if your file contains only ten characters, the operating system will allocate a full sector, there will be space left over in the sector. This is slack space. It is possible for a user to hide malicious code, tools, or clues in slack space, as well as in the free space. You may also find information in slack space from files that previously occupied that same physical sector on the drive. Therefore, an investigator should review slack space using utilities that can display the information stored in that area.

17.7 WEAKNESSES

Within the field of computer forensic science, as in any relatively young discipline, there are weaknesses to be found. In computer forensics, the main culprits are training, operational standards, and international standardization.

1. *Training:* There are many private organizations offering computer forensic seminars and classes. With the growth of computer crime, computer forensic training is a worthwhile investment for any organization—but who should receive it? Computer forensic evidence is very volatile. For preserving it, law enforcement personnel should be trained to handle it. Network operators should also be trained, to improve their abilities in intrusion detection, and lawyers should receive some training to give a basic understanding of computer evidence.

2. *Operational standard.* Computer crime, perhaps more than any other, can be international in scope. There is a need for basic guidelines for the evidence collection process to be established worldwide. This ranges from broad principles that apply to nearly every investigation, through organizational practices so that a minimum standard of planning, performance, monitoring, recording, and reporting is maintained, to recommended procedures, software, and hardware solutions.
3. *International standardization.* Different countries each have their own computer forensic methods, different standards, and their own laws. What is acceptable evidence in one country may not be acceptable in another. This is a serious problem when dealing with international crimes, as computer crime often have no boundaries. Criminals can harm the system in one country by hacking from another country. The Internet may have no boundaries, but law enforcement does. Investigations that leap from server to server, from country to country, crossing many borders on the way are complicated not only by evidence handling differences, but also by political issues and legal issues, too.

In some countries the computer networks are owned and controlled by government agencies. They may or may not cooperate with foreign governments for investigating a crime. What is considered to be hacking in India is not considered to be a crime in other countries, protecting the individual that committed the crime. Fortunately, efforts are being made to bring some standardization to procedures regarding digital evidence.

The G8 group has recommended six principles for digital evidence gathering:

1. All standard forensic and procedural principles must be applied.
2. Upon seizing digital evidence, actions taken should not alter the evidence.
3. The training should be given to the people who access the original digital evidence.
4. Complete document of all activities relating to the seizure, access, storage, or transfer of digital evidence must be kept.
5. Individuals are responsible for all actions taken while the digital evidence is in their possession.
6. Any agency that is responsible for seizing, accessing, storing, or transferring digital evidence is responsible for complying with these principles.

This is a start in standardizing computer forensic evidence gathering procedures, but there is still a long way to go. Many countries have not adopted these recommendations, and probably will not if they do not have the necessary training resources. The sting in the tail is that these are the countries that pose the greatest computer crime threat.

SUMMARY

Computer forensics is simply the application of computer investigation and analysis techniques in the interests of determining potential legal evidence. Data lost

Index

Access control, [4](#), [9](#), 162, 240, 342
matrix, 343
Add subkey, 76
step, 77
Additive identity, 147
Advanced Encryption Standard (AES), 41, 42, 68, 74, 81, 84
AFIS (Automated Fingerprint Identification System), 168
Aggregation, [295](#)
Alert protocol, 272
Anomaly detection, 283, 286
systems, 287
Anti-virus (software), 309
Application gateways, 331
Application level gateways, [330](#)
Associativity, 112
Assurance services, [10](#)
Asymmetric encryption(s), [5](#), [9](#), [12](#), [14](#), 118
Asymmetric key cryptography, 119
Attack, 285, [319](#)
Attack aggregation, [295](#)
Authentication [5](#), [6](#), 119, 121, 145, 162, 205, 219, 264, 298,
Authentication header, 241
Authentication Header (AH) protocol, 244
Authentication server, 185, 186
Authentication ticket, 191
Authentication tokens, 162
Authenticator, 187, 189
Authorization, [9](#), 163

Baby-step giant-step algorithm, 116
Back-door attack, [320](#)
Backdoors, 221

Base-rate fallacy, [296](#)
Bastion host, 331, 336, 339
BCP (Business Continuity Planning), [10](#)
Behaviour blocking, 318
Bent functions, 70
Biometric authentication method, 167
Birthday attack, 175
Block cipher(s), [14](#), [17](#), 32, 43, 74
Blowfish, 87, 89, 92, 104
Boot sector virus, [307](#)
Brute force, 95,
attack, 67, 103, 198, [320](#)
methods, 42
Bucket brigade attack, 145
Byte substitution, 76, 83
step, 78
ByteSub function, 85

Caesar cipher, [15](#)
CAST-128, 71
Certificate Authority (CA), 196, 204, 278
Certificate Revocation Lists (CRLs), 202
ChangeCipher SpecProtocol, 272
CHAP (Challenge Handshake Authentication Protocol), 214
Checksum, 172
Chinese Remainder Theorem (CRT), 111, 132
Chosen-plaintext attack, [24](#)
Cipher Block Chaining (CBC), 32, 34, 327
Cipher Feedback (CFB) mode, 35
Ciphertext, [11](#), [13](#), 120
attack, [21](#)
Circuit (proxies) gateways, [330](#)
Circuit level gateways, 333
Composite numbers, 106

- Computer
 forensics, 346
 security, 3
 security audit, 9
 virus, 306
- COMSEC (Communication Security), 10
- Confidentiality, 2, 121, 219
 and authentication, 219
- Conventional encryption, 12
- Cookie exchange, 264
- Coprime, 109, 113
- Counter mode, 38
- Cracking, 67
- Cross-realm, 199
- Cross-realm authentication, 194
- Cryptanalysis, 12, 14, 91, 96
- Cryptanalyst, 14
- Cryptanalytic attack, 72
- Cryptography, 6, 8, 12
- Cryptology, 12
- Cryptosystem, 11
- CTR (counter mode), 38
- Data encryption algorithm, 42
- Data encryption standard (DES), 42, 67, 84
- Data security, 6
- Data-hiding, 29
- Decryption, 9, 11, 94, 103
- Decryption algorithm, 13, 120
- Denial-of-service attack, 323
- DES cracker, 42, 68, 69
- DES-like ciphers, 40
- Dictionary attack, 321
- Differential cryptanalysis, 72, 95
- Diffie–Hellman encryption, 118
- Diffie–Hellman key exchange, 264
- Diffie–Hellman public key algorithm, 141
- Digital
 evidence, 358
 immune system, 318
 signature, 204, 207, 233
 signature algorithm, 177, 209
 signature standard, 209, 213
- Discrete logarithms, 114
- Distributed denial-of-service, 323
- Distributed IDS (dIDS), 293
- Distribution of public keys, 138
- DMZ (Demilitarized Zone), 332
- Documentation, 348
- DoS, 323
- Double columnar transposition, 27
- Double DES, 67
- Doubling, 148
- Dual-homed host architecture, 335
- E-mail, 355
 bombing, 325
 spamming, 325
 viruses, 308
 worms, 314
- EAP (Extensible Authentication Protocol), 170, 214
- EAP method, 170
- Eavesdropper, 143
- Electronic Code Book (ECB), 32, 33
- Electronic
 evidence, 351
 signature, 205
- ElGamal signature, 211
- Elliptic curve, 145, 146
 cryptography, 154
 cryptosystems, 156
 Diffie–Hellman, 154
 DSA, 212
 primality test, 110
- Encapsulating Security Payload (ESP), 241, 247
 header, 249
- Encryption, 5, 9, 11, 29, 94
 algorithm, 13, 119
- Euler totient function, 107
- Euler's theorem, 114, 117
- Expansion permutation, 45
- Expert systems, 289
- Extensible authentication protocol, 170
- Exterior router, 340
- External threats, 3
- Extraction, 347
- f-function, 89, 100
- Factorization, 109
- False Accept Ratio (FAR), 169
- False Reject Ratio (FRR), 169
- Fast deterministic tests, 110
- Feistel ciphers, 40
- Fermat primality test, 110
- Fermat's little theorem, 210
- Fermat's theorem, 107
- File slack, 357
- Firewall(s), 252, 329
 architectures, 334
- Free space, 357
- Gaussian primes, 106
- Governance, 9
- Greatest common divisor, 112
- Hacker, 8

- Hacking, 8
Hamiltonian cycle, 158
Handshake protocol, 272, 276
Hash, 175
 function, 40
 message authentication code, 183
 value, 173
Heuristic test, 110
Hidden directories, 29
Hiding directories, 29
Hill cipher, 20, 21
HMAC, 182
Hoax, 319
Honeypots, 297
Host-based systems, 283
- Index calculus algorithm, 114
Information security, 1, 2
Information systems security, 1
Integrity, 2, 121
 Integrity Check Value (ICV), 245
Integrity checks, 37
Inter-session chosen plaintext attacks, 198
Interior router, 339
International Data Encryption Algorithm (IDEA), 71, 101
Internet Assigned Numbers Authority (IANA), 225
Internet engineering task force, 170
Internet privacy, 6
Internet protocol spoofing, 321
Internet Security Association and Key Management Protocol (ISAKMP), 242, 253
Interpretation, 347
Initialization, 184
Intruder, 302
 detection, 302
Intrusion, 5
 detection, 281
Intrusion Detection System (IDS), 5, 282, 284, 325
Inverse shift row, 81
IP address, 329
IP security, 241
IPsec, 239
IPsec protocols, 243
IPv4, 242
IPv6, 242
IRC worms, 315
ISAKMP exchange types, 257
ISAKMP payloads, 255
Isomorphic graphs, 158
Issuer name, 200
Issuer unique identifier, 201
- Kerberos, 163, 184
 authentication, 186, 190
 authentication model, 192
Key, 13
 addition transform, 75
 distribution servers, 190
 escrow, 222
 establishment protocol, 154
 exchange payload, 256
 expansion, 94
 generation algorithm, 126
 length, 128
 size, 21, 85
Keystroke monitoring, 289
- Linear cryptanalysis, 72
Linear mix transform, 75
Long-term keys, 186
Lucas–Lehmer test, 111
- Macro viruses, 307
Malicious programs, 306
Malware, 5, 306
Man-in-the-middle attack(s), 144, 145, 167, 321, 324
Masquerader, 282
MD4, 172, 173
MD5, 25, 171, 173, 174
Meet-in-the-middle attack, 67
Memory resident virus, 307
Message Authentication Code (MAC), 183, 245
Message digest, 177, 206
Message Digest Algorithm 2 (MD2), 172
Message
 encryption, 233
 padding, 179
Miller–Rabin primality test, 110
Miller–Rabin test, 111
MIME (Secure/Multipurpose Internet Mail Extensions), 223, 224
Misfeasor, 282
Misuse detection, 283, 286
 systems, 288
Mix column, 76, 82, 85
 step, 81
Model based intrusion detection, 289
Modes of operation, 32
Monoalphabetic cipher(s), 15, 16, 18
Montgomery algorithm, 132
- Naïve methods, 109
Network security, 7

- Network Security Monitor (NSM)**, [293](#)
Network-based system, [283](#)
NIDS, [283](#)
NIST (National Institute of Standards and Technology), [74](#), [156](#), [209](#)
Non-repudiation, [9](#), [206](#)
- OAKLEY**, [263](#)
OAKLEY key determination protocol, [242](#), [263](#)
One Time Password (OTP), [171](#)
One-time pad, [24](#)
One-way, [28](#)
One-way hash value, [28](#)
Operational model, [7](#)
Output Feedback (OFB) mode, [37](#)
- P-box permutation**, [45](#)
Packet filtering, [336](#)
Packet filters, [330](#)
Packet filter firewalls, [330](#)
Padding, [172](#), [175](#), [183](#)
PAP (Password Authentication Protocol), [214](#)
Parasitic viruses, [307](#)
Passive system, [283](#)
Password-based authentication, [164](#)
Passwords, [298](#)
Pattern matching, [291](#)
Penetration, [285](#)
Perimeter network, [338](#)
Permutation, [16](#)

PGP, [222](#)
Phishing, [322](#)
PKIs, [196](#), [205](#), [267](#)
Plaintext, [11](#), [12](#), [119](#)
Playfair cipher, [17](#), [18](#)
Point at infinity, [150](#)
Point-to-point protocol, [170](#)
Polyalphabetic ciphers, [22](#)

Polygraphic substitution cipher, [19](#)
Polymorphic code, [312](#)
 virus, [307](#)
Predictive pattern generation, [288](#)
Preservation, [347](#)
Pretty Good Privacy (PGP), [138](#), [216](#)
Primality test, [108](#), [117](#)
Prime number, [106](#)
Private key, [119](#)
Probabilistic tests, [109](#)
Production honeypots, [297](#)
- Proxy**
 gateways, [331](#)
 server, [331](#)
Pseudo-random generation algorithm, [98](#)
Pseudo-random number, [24](#)
Public announcement (method of distribution), [138](#)
Public key, [118](#)
 authority, [139](#)
 certificates, [140](#)
 cryptosystems, [123](#)
 encryption, [14](#)
 infrastructure, [138](#), [204](#)
 and private key, [119](#)
Publicly available directory, [139](#)
- RADIUS**, [214](#)
Randomized primality tests, [109](#)
Raw RC5 block cipher, [94](#)
RC4, [95](#), [97](#), [98](#)
RC5, [92](#), [98](#)
RC5-CBC Pad, [95](#)
RC5-CBC, [94](#)
RC5-CTS cipher, [95](#)
RC6, [98](#), [99](#), [100](#)
Reactive system, [283](#)
Reliability, [10](#)
Remote TGS, [195](#)
Replay attacks, [198](#)
Research honeypots, [298](#)
Rijndael, [84](#)
Rijndael algorithm, [74](#)
Rijndael's inverse S-box, [80](#)
RIPEMD-160, [175](#), [183](#)
Risk, [285](#)
Risk assessment, [9](#)
ROC (Receiver Operating Characteristic), [169](#)
ROISI, [9](#)
Round key addition, [85](#)
Round keys, [87](#)
RSA, [95](#), [125](#)
 blinding, [133](#)
 public key encryption, [124](#)
Rule-based Intrusion Detection (RBID), [292](#)
- S-box(es)**, [42](#), [70](#), [79](#), [80](#), [84](#), [87](#), [88](#)
S-box substitution, [45](#)
S/MIME, [232](#)
Sandboxing, [318](#)
Screened host architecture, [336](#)
Screened subnet, [337](#)

Cryptography and Information Security

V.K. Pachghare

This well-organized text presents the principles, techniques, design, and implementation of cryptography and information security algorithms, with a perfect balance in the presentation of theoretical and practical aspects. To provide the mathematical background required to understand the principles of cryptography and information security, the text explains all the relevant theorems such as Fermat's theorem and Euler's theorem. The book gives a clear analysis of various encryption methods and cipher techniques. In addition, various security measures, for example, firewalls and virtual private network, and web security, are also discussed.

KEY FEATURES

- Covers the latest topic of computer forensics and the areas in which they can be applied.
- Gives algorithms with numerical explanations.
- Provides a large number of solved problems.

The book is intended for the undergraduate and postgraduate students of computer science and engineering (B.Tech./M.Tech.), undergraduate and postgraduate students of computer science (B.Sc./M.Sc. Computer Science), and information technology (B.Sc./M.Sc. IT) and the students of Master of Computer Applications (MCA).

THE AUTHOR

V.K. PACHGHARE is Assistant Professor, Department of Computer Engineering and Information Technology, Government College of Engineering, Pune. He has seventeen years of teaching experience and has published a book on computer graphics.

You may also be interested in

Digital Signature: Network Security Practices, Kailash N. Gupta, Kamlesh N. Agarwala & Prateek A. Agarwala

Information Security: Theory and Practice, Dhiren R. Patel

Microsoft Encyclopedia of Security, Tulloch

Microsoft Windows Server 2003 PKI and Certificate Security, Komar with the Microsoft PKI Team

Network Security and Management, Brijendra Singh

Windows Server 2008 PKI and Certificate Security, Komar

Look Both Ways: Help Protect Your Family on the Internet, Criddle

Rs. 275.00

www.phindia.com

ISBN: 978-81-203-3521-9



9 788120 335219