

大学计算机教育国外著名教材系列



Introduction to Computer Security

计算机安全导论



Michael T. Goodrich 著
Roberto Tamassia

清华大学出版社

大学计算机教育国外著名教材系列（影印版）

Introduction to Computer Security

计算机安全导论

Michael T. Goodrich

Roberto Tamassia

著

清华大学出版社
北京

English reprint edition copyright © 2012 by PEARSON EDUCATION ASIA LIMITED and TSINGHUA UNIVERSITY PRESS.

Original English language title from Proprietor's edition of the Work.

Original English language title: Introduction to Computer Security by Michael T. Goodrich, Roberto Tamassia, Copyright © 2012
All Rights Reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Prentice Hall, Inc.

This edition is authorized for sale and distribution only in the People's Republic of China (excluding the Special Administrative Region of Hong Kong, Macao SAR and Taiwan).

本书影印版由 Pearson Education (培生教育出版集团) 授权给清华大学出版社出版发行。

For sale and distribution in the People's Republic of China exclusively (except Taiwan, Hong Kong SAR and Macao SAR).

仅限于中华人民共和国境内(不包括中国香港、澳门特别行政区和中国台湾地区)销售发行。

北京市版权局著作权合同登记号 图字 01-2012-4367 号

本书封面贴有 Pearson Education(培生教育出版集团)激光防伪标签, 无标签者不得销售。
版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

计算机安全导论 = Introduction to Computer Security: 英文/(美)古德里奇(Goodrich, M. T.), (美)塔玛萨(Tamassia, R.)著.--影印本.--北京:清华大学出版社, 2013.1

大学计算机教育国外著名教材系列(影印版)

ISBN 978-7-302-30719-8

I. ①计… II. ①古… ②塔… III. ①计算机安全—高等学校—教材—英文 IV. ①TP309

中国版本图书馆 CIP 数据核字(2012)第 278567 号

责任编辑: 龙启铭

封面设计: 张海清

责任印制: 何 芊

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社 总 机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者: 北京四季青印刷厂

装 订 者: 三河市溧源装订厂

发 行 者: 全国新华书店

开 本: 185×230 印张: 36

版 次: 2013 年 1 月第 1 版

印 次: 2013 年 1 月第 1 次印刷

印 数: 1~3000

定 价: 79.00 元

产品编号: 040262-01

Contents

1	Introduction	1
1.1	Fundamental Concepts	2
1.1.1	Confidentiality, Integrity, and Availability	3
1.1.2	Assurance, Authenticity, and Anonymity	9
1.1.3	Threats and Attacks	14
1.1.4	Security Principles	15
1.2	Access Control Models	19
1.2.1	Access Control Matrices	19
1.2.2	Access Control Lists	20
1.2.3	Capabilities	22
1.2.4	Role-Based Access Control	23
1.3	Cryptographic Concepts	25
1.3.1	Encryption	25
1.3.2	Digital Signatures	31
1.3.3	Simple Attacks on Cryptosystems	32
1.3.4	Cryptographic Hash Functions	35
1.3.5	Digital Certificates	37
1.4	Implementation and Usability Issues	39
1.4.1	Efficiency and Usability	39
1.4.2	Passwords	41
1.4.3	Social Engineering	43
1.4.4	Vulnerabilities from Programming Errors	44
1.5	Exercises	46

2	Physical Security	55
2.1	Physical Protections and Attacks	56
2.2	Locks and Safes	57
2.2.1	Lock Technology	57
2.2.2	Attacks on Locks and Safes	62
2.2.3	The Mathematics of Lock Security	68
2.3	Authentication Technologies	71
2.3.1	Barcodes	71
2.3.2	Magnetic Stripe Cards	72
2.3.3	Smart Cards	74
2.3.4	RFIDs	79
2.3.5	Biometrics	83
2.4	Direct Attacks Against Computers	88
2.4.1	Environmental Attacks and Accidents	88
2.4.2	Eavesdropping	89
2.4.3	TEMPEST	94
2.4.4	Live CDs	96
2.4.5	Computer Forensics	96
2.5	Special-Purpose Machines	99
2.5.1	Automated Teller Machines	99
2.5.2	Voting Machines	101
2.6	Physical Intrusion Detection	103
2.6.1	Video Monitoring	103
2.6.2	Human Factors and Social Engineering	105
2.7	Exercises	106
3	Operating Systems Security	113
3.1	Operating Systems Concepts	114
3.1.1	The Kernel and Input/Output	115
3.1.2	Processes	116
3.1.3	The Filesystem	121
3.1.4	Memory Management	124
3.1.5	Virtual Machines	128
3.2	Process Security	130
3.2.1	Inductive Trust from Start to Finish	130
3.2.2	Monitoring, Management, and Logging	132
3.3	Memory and Filesystem Security	136
3.3.1	Virtual Memory Security	136
3.3.2	Password-Based Authentication	137
3.3.3	Access Control and Advanced File Permissions	140
3.3.4	File Descriptors	146
3.3.5	Symbolic Links and Shortcuts	148

3.4	Application Program Security	149
3.4.1	Compiling and Linking	149
3.4.2	Simple Buffer Overflow Attacks	150
3.4.3	Stack-Based Buffer Overflow	152
3.4.4	Heap-Based Buffer Overflow Attacks	159
3.4.5	Format String Attacks	162
3.4.6	Race Conditions	163
3.5	Exercises	166
4	Malware	173
4.1	Insider Attacks	174
4.1.1	Backdoors	174
4.1.2	Logic Bombs	177
4.1.3	Defenses Against Insider Attacks	180
4.2	Computer Viruses	181
4.2.1	Virus Classification	182
4.2.2	Defenses Against Viruses	185
4.2.3	Encrypted Viruses	186
4.2.4	Polymorphic and Metamorphic Viruses	187
4.3	Malware Attacks	188
4.3.1	Trojan Horses	188
4.3.2	Computer Worms	190
4.3.3	Rootkits	195
4.3.4	Zero-Day Attacks	199
4.3.5	Botnets	200
4.4	Privacy-Invasive Software	202
4.4.1	Adware	202
4.4.2	Spyware	204
4.5	Countermeasures	208
4.5.1	Best Practices	208
4.5.2	The Impossibility of Detecting All Malware	211
4.5.3	The Malware Detection Arms Race	213
4.5.4	Economics of Malware	214
4.6	Exercises	215
5	Network Security I	221
5.1	Network Security Concepts	222
5.1.1	Network Topology	222
5.1.2	Internet Protocol Layers	223
5.1.3	Network Security Issues	227
5.2	The Link Layer	229
5.2.1	Ethernet	229

5.2.2	Media Access Control (MAC) Addresses	232
5.2.3	ARP Spoofing	233
5.3	The Network Layer	236
5.3.1	IP	236
5.3.2	Internet Control Message Protocol	240
5.3.3	IP Spoofing	242
5.3.4	Packet Sniffing	244
5.4	The Transport Layer	246
5.4.1	Transmission Control Protocol (TCP)	246
5.4.2	User Datagram Protocol (UDP)	250
5.4.3	Network Address Translation (NAT)	251
5.4.4	TCP Session Hijacking	253
5.5	Denial-of-Service Attacks	256
5.5.1	ICMP Attacks	256
5.5.2	SYN Flood Attacks	258
5.5.3	Optimistic TCP ACK Attack	260
5.5.4	Distributed Denial-of-Service	261
5.5.5	IP Traceback	262
5.6	Exercises	264
6	Network Security II	269
6.1	The Application Layer and DNS	270
6.1.1	A Sample of Application-Layer Protocols	270
6.1.2	The Domain Name System (DNS)	271
6.1.3	DNS Attacks	278
6.1.4	DNSSEC	285
6.2	Firewalls	287
6.2.1	Firewall Policies	288
6.2.2	Stateless and Stateful Firewalls	289
6.3	Tunneling	292
6.3.1	Secure Shell (SSH)	293
6.3.2	IPsec	294
6.3.3	Virtual Private Networking (VPN)	297
6.4	Intrusion Detection	299
6.4.1	Intrusion Detection Events	302
6.4.2	Rule-Based Intrusion Detection	305
6.4.3	Statistical Intrusion Detection	306
6.4.4	Port Scanning	308
6.4.5	Honeypots	312
6.5	Wireless Networking	313
6.5.1	Wireless Technologies	314
6.5.2	Wired Equivalent Privacy (WEP)	315

6.5.3	Wi-Fi Protected Access (WPA)	318
6.6	Exercises	322
7	Web Security	327
7.1	The World Wide Web	328
7.1.1	HTTP and HTML	328
7.1.2	HTTPS	334
7.1.3	Dynamic Content	339
7.1.4	Sessions and Cookies	342
7.2	Attacks on Clients	347
7.2.1	Session Hijacking	347
7.2.2	Phishing	349
7.2.3	Click-Jacking	351
7.2.4	Vulnerabilities in Media Content	352
7.2.5	Privacy Attacks	356
7.2.6	Cross-Site Scripting (XSS)	357
7.2.7	Cross-Site Request Forgery (CSRF)	364
7.2.8	Defenses Against Client-Side Attacks	366
7.3	Attacks on Servers	368
7.3.1	Server-Side Scripting	368
7.3.2	Server-Side Script Inclusion Vulnerabilities	370
7.3.3	Databases and SQL Injection Attacks	372
7.3.4	Denial-of-Service Attacks	378
7.3.5	Web Server Privileges	379
7.3.6	Defenses Against Server-Side Attacks	380
7.4	Exercises	382
8	Cryptography	387
8.1	Symmetric Cryptography	388
8.1.1	Attacks	389
8.1.2	Substitution Ciphers	391
8.1.3	One-Time Pads	393
8.1.4	Pseudo-Random Number Generators	395
8.1.5	The Hill Cipher and Transposition Ciphers	397
8.1.6	The Advanced Encryption Standard (AES)	399
8.1.7	Modes of Operation	402
8.2	Public-Key Cryptography	406
8.2.1	Modular Arithmetic	406
8.2.2	The RSA Cryptosystem	410
8.2.3	The Elgamal Cryptosystem	413
8.2.4	Key Exchange	415
8.3	Cryptographic Hash Functions	417

8.3.1	Properties and Applications	417
8.3.2	Birthday Attacks	419
8.4	Digital Signatures	421
8.4.1	The RSA Signature Scheme	422
8.4.2	The Elgamal Signature Scheme	423
8.4.3	Using Hash Functions with Digital Signatures	424
8.5	Details of AES and RSA Cryptography	425
8.5.1	Details for AES	425
8.5.2	Details for RSA	431
8.6	Exercises	439
9	Security Models and Practice	445
9.1	Policy, Models, and Trust	446
9.1.1	Security Policy	446
9.1.2	Security Models	447
9.1.3	Trust Management	448
9.2	Access-Control Models	450
9.2.1	The Bell-La Padula Model	450
9.2.2	Other Access-Control Models	454
9.2.3	Role-Based Access Control	456
9.3	Security Standards and Evaluation	460
9.3.1	Orange Book and Common Criteria	460
9.3.2	Government Regulations and Standards	462
9.4	Software Vulnerability Assessment	464
9.4.1	Static and Dynamic Analysis	465
9.4.2	Exploit Development and Vulnerability Disclosure	468
9.5	Administration and Auditing	470
9.5.1	System Administration	470
9.5.2	Network Auditing and Penetration Testing	473
9.6	Kerberos	475
9.6.1	Kerberos Tickets and Servers	475
9.6.2	Kerberos Authentication	476
9.7	Secure Storage	479
9.7.1	File Encryption	479
9.7.2	Disk Encryption	481
9.7.3	Trusted Platform Module	482
9.8	Exercises	484
10	Distributed-Applications Security	487
10.1	Database Security	488
10.1.1	Tables and Queries	489
10.1.2	Updates and the Two-Phase Commit Protocol	491

10.1.3 Database Access Control	493
10.1.4 Sensitive Data	497
10.2 Email Security	500
10.2.1 How Email Works	500
10.2.2 Encryption and Authentication	502
10.2.3 Spam	507
10.3 Payment Systems and Auctions	513
10.3.1 Credit Cards	513
10.3.2 Digital Cash	516
10.3.3 Online Auctions	518
10.4 Digital-Rights Management	519
10.4.1 Digital-Media Rights Techniques	520
10.4.2 Digital-Media Rights Practice	523
10.4.3 Software Licensing Schemes	525
10.4.4 Legal Issues	527
10.5 Social Networking	528
10.5.1 Social Networks as Attack Vectors	528
10.5.2 Privacy	529
10.6 Voting Systems	531
10.6.1 Security Goals	531
10.6.2 ThreeBallot	532
10.7 Exercises	535

Chapter 1

Introduction

Contents

1.1 Fundamental Concepts	2
1.1.1 Confidentiality, Integrity, and Availability	3
1.1.2 Assurance, Authenticity, and Anonymity	9
1.1.3 Threats and Attacks	14
1.1.4 Security Principles	15
1.2 Access Control Models	19
1.2.1 Access Control Matrices	19
1.2.2 Access Control Lists	20
1.2.3 Capabilities	22
1.2.4 Role-Based Access Control	23
1.3 Cryptographic Concepts	25
1.3.1 Encryption	25
1.3.2 Digital Signatures	31
1.3.3 Simple Attacks on Cryptosystems	32
1.3.4 Cryptographic Hash Functions	35
1.3.5 Digital Certificates	37
1.4 Implementation and Usability Issues	39
1.4.1 Efficiency and Usability	39
1.4.2 Passwords	41
1.4.3 Social Engineering	43
1.4.4 Vulnerabilities from Programming Errors	44
1.5 Exercises	46

1.1 Fundamental Concepts

In this chapter, we introduce several fundamental concepts in computer security. Topics range from theoretical cryptographic primitives, such as digital signatures, to practical usability issues, such as social engineering. This chapter provides an informal and intuitive description of a variety of topics that will be covered in more detail in the rest of the book.

Existing computer systems may contain legacy features of earlier versions dating back to bygone eras, such as when the Internet was the sole domain of academic researchers and military labs. For instance, assumptions of trust and lack of malicious behavior among network-connected machines, which may have been justifiable in the early eighties, are surprisingly still present in the way the Internet operates today. Such assumptions have led to the growth of Internet-based crime.

An important aspect of computer security is the identification of *vulnerabilities* in computer systems, which can, for instance, allow a malicious user to gain access to private data and even assume full control of a machine. Vulnerabilities enable a variety of *attacks*. Analysis of these attacks can determine the severity of damage that can be inflicted and the likelihood that the attack can be further replicated. Actions that need to be taken to defend against attacks include identifying compromised machines, removing the malicious code, and patching systems to eliminate the vulnerability.

In order to have a secure computer system, sound *models* are a first step. In particular, it is important to define the *security properties* that must be assured, anticipate the types of *attacks* that could be launched, and develop specific defenses. The *design* should also take into account usability issues. Indeed, security measures that are difficult to understand and inconvenient to follow will likely lead to failure of adoption. Next, the hardware and software *implementation* of a system needs to be rigorously *tested* to detect programming errors that introduce vulnerabilities. Once the system is deployed, procedures should be put in place to *monitor* the behavior of the system, detect security breaches, and react to them. Finally, security-related *patches* to the system must be applied as soon as they become available.

Computer security concepts often are better understood by looking at issues in a broader context. For this reason, this book also includes discussions of the security of various physical and real-world systems, including locks, ATM machines, and passenger screening at airports.

1.1.1 Confidentiality, Integrity, and Availability

Computers and networks are being misused at a growing rate. Spam, phishing, and computer viruses are becoming multibillion-dollar problems, as is identity theft, which poses a serious threat to the personal finances and credit ratings of users, and creates liabilities for corporations. Thus, there is a growing need for broader knowledge of computer security in society as well as increased expertise among information technology professionals. Society needs more security-educated computer professionals, who can successfully defend against and prevent computer attacks, as well as security-educated computer users, who can safely manage their own information and the systems they use.

One of the first things we need to do in a book on computer security is to define our concepts and terms. Classically, information security has been defined in terms of the acronym *C.I.A.*, which in this case stands for *confidentiality, integrity, and availability*. (See Figure 1.1.)

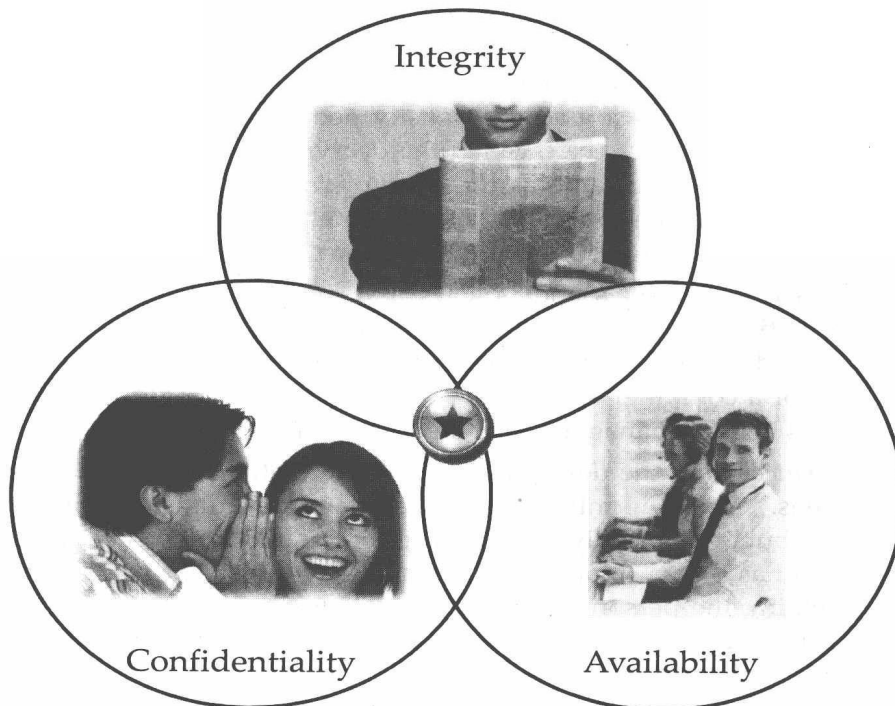


Figure 1.1: The C.I.A. concepts: confidentiality, integrity, and availability.

Confidentiality

In the context of computer security, *confidentiality* is the avoidance of the unauthorized disclosure of information. That is, confidentiality involves the protection of data, providing access for those who are allowed to see it while disallowing others from learning anything about its content.

Keeping information secret is often at the heart of information security, and this concept, in fact, predates computers. For example, in the first recorded use of cryptography, Julius Caesar communicated commands to his generals using a simple cipher. In his cipher, Caesar took each letter in his message and substituted D for A, E for B, and so on. This cipher can be easily broken, making it an inappropriate tool for achieving confidentiality today. But in its time, the Caesar cipher was probably fairly secure, since most of Caesar's enemies couldn't read Latin anyway.

Nowadays, achieving confidentiality is more of a challenge. Computers are everywhere, and each one is capable of performing operations that could compromise confidentiality. With all of these threats to the confidentiality of information, computer security researchers and system designers have come up with a number of tools for protecting sensitive information. These tools incorporate the following concepts:

- **Encryption:** the transformation of information using a secret, called an encryption key, so that the transformed information can only be read using another secret, called the decryption key (which may, in some cases, be the same as the encryption key). To be secure, an encryption scheme should make it extremely difficult for someone to determine the original information without use of the decryption key.
- **Access control:** rules and policies that limit access to confidential information to those people and/or systems with a "need to know." This need to know may be determined by identity, such as a person's name or a computer's serial number, or by a role that a person has, such as being a manager or a computer security specialist.
- **Authentication:** the determination of the identity or role that someone has. This determination can be done in a number of different ways, but it is usually based on a combination of something the person has (like a smart card or a radio key fob storing secret keys), something the person knows (like a password), and something the person is (like a human with a fingerprint). The concept of authentication is schematically illustrated in Figure 1.2.
- **Authorization:** the determination if a person or system is allowed access to resources, based on an access control policy. Such authorizations should prevent an attacker from tricking the system into letting him have access to protected resources.

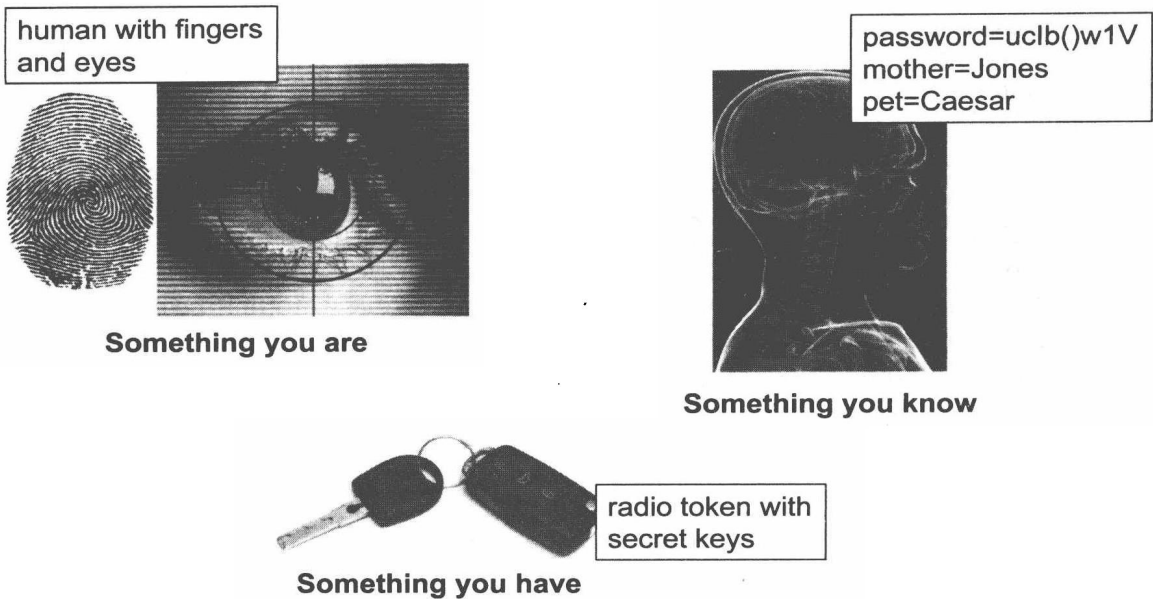


Figure 1.2: Three foundations for authentication.

- **Physical security:** the establishment of physical barriers to limit access to protected computational resources. Such barriers include locks on cabinets and doors, the placement of computers in windowless rooms, the use of sound dampening materials, and even the construction of buildings or rooms with walls incorporating copper meshes (called *Faraday cages*) so that electromagnetic signals cannot enter or exit the enclosure.

When we visit a web page that asks for our credit card number and our Internet browser shows a little lock icon in the corner, there is a lot that has gone on in the background to help ensure the confidentiality of our credit card number. In fact, a number of tools have probably been brought to bear here. Our browser begins the process by performing an authentication procedure to verify that the web site we are connecting to is indeed who it says it is. While this is going on, the web site might itself be checking that our browser is authentic and that we have the appropriate authorizations to access this web page according to its access control policy. Our browser then asks the web site for an encryption key to encrypt our credit card, which it then uses so that it only sends our credit card information in encrypted form. Finally, once our credit card number reaches the server that is providing this web site, the data center where

the server is located should have appropriate levels of physical security, access policies, and authorization and authentication mechanisms to keep our credit card number safe. We discuss these topics in some detail in this book.

For instance, in Section 2.4.2, we study a number of real demonstrated risks to physical eavesdropping. For example, researchers have shown that one can determine what someone is typing just by listening to a recording of their key strokes. Likewise, experiments show that it is possible to reconstruct the image of a computer screen either by monitoring its electromagnetic radiation or even from a video of a blank wall that the screen is shining on. Thus, physical security is an information security concept that should not be taken for granted.

Integrity

Another important aspect of information security is *integrity*, which is the property that information has not be altered in an unauthorized way.

The importance of integrity is often demonstrated to school children in the *Telephone game*. In this game, a group of children sit in a circle and the person who is “it” whispers a message in the ear of his or her neighbor on the right. Each child in the circle then waits to listen to the message from his or her neighbor on the left. Once a child has received the message, he or she then whispers this same message to their neighbor on the right. This message passing process continues until the message goes full circle and returns to the person who is “it.” At that point, the last person to hear the message says the message out loud so that everyone can hear it. Typically, the message has been so mangled by this point that it is a great joke to all the children, and the game is repeated with a new person being “it.” And, with each repeat play, the game reinforces that this whispering process rarely ever preserves data integrity. Indeed, could this be one of the reasons we often refer to rumors as being “whispered”?

There are a number of ways that data integrity can be compromised in computer systems and networks, and these compromises can be benign or malicious. For example, a benign compromise might come from a storage device being hit with a stray cosmic ray that flips a bit in an important file, or a disk drive might simply crash, completely destroying some of its files. A malicious compromise might come from a computer virus that infects our system and deliberately changes some the files of our operating system, so that our computer then works to replicate the virus and send it to other computers. Thus, it is important that computer systems provide tools to support data integrity.

The previously mentioned tools for protecting the confidentiality of information, denying access to data to users without appropriate access rights, also help prevent data from being modified in the first place. In addition, there are several tools specifically designed to support integrity, including the following:

- **Backups:** the periodic archiving of data. This archiving is done so that data files can be restored should they ever be altered in an unauthorized or unintended way.
- **Checksums:** the computation of a function that maps the contents of a file to a numerical value. A checksum function depends on the entire contents of a file and is designed in a way that even a small change to the input file (such as flipping a single bit) is highly likely to result in a different output value. Checksums are like trip-wires—they are used to detect when a breach to data integrity has occurred.
- **Data correcting codes:** methods for storing data in such a way that small changes can be easily detected and automatically corrected. These codes are typically applied to small units of storage (e.g., at the byte level or memory word level), but there are also data-correcting codes that can be applied to entire files as well.

These tools for achieving data integrity all possess a common trait—they use *redundancy*. That is, they involve the replication of some information content or functions of the data so that we can detect and sometimes even correct breaches in data integrity.

In addition, we should stress that it is not just the content of a data file that needs to be maintained with respect to integrity. We also need to protect the *metadata* for each data file, which are attributes of the file or information about access to the file that are not strictly a part of its content. Examples of metadata include the user who is the owner of the file, the last user who has modified the file, the last user who has read the file, the dates and times when the file was created and last modified and accessed, the name and location of the file in the file system, and the list of users or groups who can read or write the file. Thus, changing any metadata of a file should be considered a violation of its integrity.

For example, a computer intruder might not actually modify the content of any user files in a system he has infiltrated, but he may nevertheless be modifying metadata, such as access time stamps, by looking at our files (and thereby compromising their confidentiality if they are not encrypted). Indeed, if our system has integrity checks in place for this type of metadata, it may be able to detect an intrusion that would have otherwise gone unnoticed.