

Microsoft®

CISSP



David R. Miller

Training Kit

Certified Information System Security Professional (CISSP) exam objective map

OBJECTIVE	CHAPTER
1.0 ACCESS CONTROL	
1.1 Control access by applying the following concepts/methodologies/ techniques	2, 3, 4, 5, 7, 10
1.1.1 Policies	1, 2, 4
1.1.2 Types of controls (preventive, detective, corrective, etc.)	2, 4, 5, 10
1.1.3 Techniques (e.g., non-discretionary, discretionary and mandatory)	2, 5
1.1.4 Identification and Authentication	2, 4, 7, 10
1.1.5 Decentralized/distributed access control techniques	2, 5, 7, 10
1.1.6 Authorization mechanisms	2, 3, 4, 5, 7, 10
1.1.7 Logging and monitoring	2, 4, 7, 9, 10
1.2 Understand access control attacks	2, 4, 9, 10
1.2.1 Threat modeling	2, 4, 5, 6, 7, 8, 9, 10
1.2.2 Asset valuation	2, 8
1.2.3 Vulnerability analysis	2, 3, 4, 5, 7, 8, 9, 10
1.2.4 Access aggregation	2, 10
1.3 Assess effectiveness of access controls	2, 4, 5, 6, 8, 9
1.3.1 User entitlement	1, 2, 4, 5, 6, 8, 10
1.3.2 Access review & audit	1, 2, 4, 5, 6, 7, 8, 9, 10
1.4 Identity and access provisioning lifecycle (e.g., provisioning, review, revocation)	1, 2, 4, 5, 10
2.0 TELECOMMUNICATIONS AND NETWORK SECURITY	
2.1 Understand secure network architecture and design (e.g., IP & non-IP protocols, segmentation)	5, 7, 8
2.1.1 OSI and TCP/IP models	7
2.1.2 IP networking	7
2.1.3 Implications of multi-layer protocols	7
2.2 Securing network components	4, 5, 7, 8, 10
2.2.1 Hardware (e.g., modems, switches, routers, wireless access points)	2, 4, 7, 8, 10
2.2.2 Transmission media (e.g., wired, wireless, fiber)	2, 3, 4, 7, 8, 10
2.2.3 Network access control devices (e.g., firewalls, proxies)	2, 4, 7, 8, 10
2.2.4 End-point security	2, 3, 4, 5, 7, 8, 10
2.3 Establish secure communication channels (e.g., VPN, TLS/SSL, VLAN)	3, 7
2.3.1 Voice (e.g., POTS, PBX, VoIP)	7
2.3.2 Multimedia collaboration (e.g., remote meeting technology, instant messaging)	7
2.3.3 Remote access (e.g., screen scraper, virtual application/desktop, telecommuting)	2, 7, 10
2.3.4 Data communications	2, 3, 5, 6, 7, 10
2.4 Understand network attacks (e.g., DDoS, spoofing)	3, 7, 8, 9, 10

Exam Objectives The exam objectives listed here are current as of this book's publication date. Exam objectives are subject to change at any time without prior notice and at the sole discretion of ISC². Please visit the ISC² Certifications webpage for the most current listing of exam objectives at <https://www.isc2.org/cissp/default.aspx>.

OBJECTIVE	CHAPTER
3.0 INFORMATION SECURITY GOVERNANCE & RISK MANAGEMENT	
3.1 Understand and align security function to goals, mission and objectives of the organization	1, 8
3.2 Understand and apply security governance	1, 2, 4, 5, 6, 8, 9, 10
3.2.1 Organizational processes (e.g., acquisitions, divestitures, governance committees)	1, 6, 8
3.2.2 Security roles and responsibilities	1, 2, 4, 6, 8, 9, 10
3.2.3 Legislative and regulatory compliance	1, 5, 6, 8
3.2.4 Privacy requirements compliance	1, 5, 6, 8, 9
3.2.5 Control frameworks	1, 2, 5, 6, 9
3.2.6 Due care	1, 5, 6, 8
3.2.7 Due diligence	1, 5, 6, 8
3.3 Understand and apply concepts of confidentiality, integrity and availability	1, 2, 3, 4, 5, 7
3.4 Develop and implement security policy	1, 5, 6, 8, 10
3.4.1 Security policies	1, 5, 6, 8
3.4.2 Standards/baselines	1, 5, 6, 8
3.4.3 Procedures	1, 5, 6, 8
3.4.4 Guidelines	1, 5, 6, 8
3.4.5 Documentation	1, 5, 6, 8, 10
3.5 Manage the information life cycle (e.g., classification, categorization, and ownership)	1, 6, 8, 9, 10
3.6 Manage third-party governance (e.g., on-site assessment, document exchange and review, process/policy review)	1, 5, 6, 8, 9, 10
3.7 Understand and apply risk management concepts	1, 5, 6, 8, 9, 10
3.7.1 Identify threats and vulnerabilities	1, 2, 4, 5, 6, 7, 8, 9, 10
3.7.2 Risk assessment/analysis (qualitative, quantitative, hybrid)	1, 2, 4, 5, 6, 8, 10
3.7.3 Risk assignment/acceptance	1, 6, 8
3.7.4 Countermeasure selection	1, 2, 3, 4, 5, 6, 7, 8, 10
3.7.5 Tangible and intangible asset valuation	1, 8
3.8 Manage personnel security	1, 4, 8, 10
3.8.1 Employment candidate screening (e.g., reference checks, education verification)	1
3.8.2 Employment agreements and policies	1, 4, 6, 8
3.8.3 Employee termination processes	1
3.8.4 Vendor, consultant and contractor controls	1, 6, 8
3.9 Develop and manage security education, training and awareness	1, 2, 3, 4, 6, 7, 8, 10
3.10 Manage the Security Function	1, 4, 5, 6, 8, 9, 10
3.10.1 Budget	1, 4, 6, 8
3.10.2 Metrics	1, 4, 5, 6, 7, 8, 9, 10
3.10.3 Resources	1, 4, 5, 6, 7, 8, 9, 10
3.10.4 Develop and implement information security strategies	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
3.10.5 Assess the completeness and effectiveness of the security program	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
4.0 SOFTWARE DEVELOPMENT SECURITY	
4.1 Understand and apply security in the software development life cycle	9
4.1.1 Development Life Cycle	9
4.1.2 Maturity models	5, 9
4.1.3 Operation and maintenance	9, 10
4.1.4 Change management	9, 10
4.2 Understand the environment and security controls	2, 4, 5, 7, 8, 9, 10
4.2.1 Security of the software environment	2, 5, 7, 8, 9
4.2.2 Security issues of programming languages	9
4.2.3 Security issues in source code (e.g., buffer overflow, escalation of privilege, backdoor)	7, 8, 9, 10
4.2.4 Configuration management	4, 8, 9, 10
4.3 Assess the effectiveness of software security	7, 8, 9, 10

OBJECTIVE	CHAPTER
5.0 CRYPTOGRAPHY	
5.1 Understand the application and use of cryptography	2, 3
5.1.1 Data at rest (e.g., Hard Drive)	1, 2, 3, 7
5.1.2 Data in transit (e.g., On the wire)	1, 2, 3, 7
5.2 Understand the cryptographic life cycle (e.g., cryptographic limitations, algorithm/protocol governance)	3
5.3 Understand encryption concepts	3
5.3.1 Foundational concepts	3
5.3.2 Symmetric cryptography	3
5.3.3 Asymmetric cryptography	3
5.3.4 Hybrid cryptography	3
5.3.5 Message digests	3
5.3.6 Hashing	3
5.4 Understand key management processes	2, 3, 7
5.4.1 Creation/distribution	2, 3, 7
5.4.2 Storage/destruction	2, 3
5.4.3 Recovery	3
5.4.4 Key escrow	3
5.5 Understand digital signatures	3
5.6 Understand non-repudiation	3
5.7 Understand methods of cryptanalytic attacks	3
5.7.1 Chosen plain-text	3
5.7.2 Social engineering for key discovery	3
5.7.3 Brute Force (e.g., rainbow tables, specialized/scalable architecture)	3
5.7.4 Cipher-text only	3
5.7.5 Known plaintext	3
5.7.6 Frequency analysis	3
5.7.7 Chosen cipher-text	3
5.7.8 Implementation attacks	3
5.8 Use cryptography to maintain network security	2, 3, 7
5.9 Use cryptography to maintain application security	3, 9
5.10 Understand Public Key Infrastructure (PKI)	3, 7
5.11 Understand certificate related issues	3
5.12 Understand information hiding alternatives (e.g., steganography, watermarking)	3
6.0 SECURITY ARCHITECTURE & DESIGN	
6.1 Understand the fundamental concepts of security models (e.g., Confidentiality, Integrity, and Multi-level Models)	2, 5
6.2 Understand the components of information systems security evaluation models	5
6.2.1 Product evaluation models (e.g., common criteria)	5
6.2.2 Industry and international security implementation guidelines (e.g., PCI-DSS, ISO)	2, 5
6.3 Understand security capabilities of information systems (e.g., memory protection, virtualization, trusted platform module)	1, 2, 3, 5, 9, 10
6.4 Understand the vulnerabilities of security architectures	1, 2, 5, 7, 8, 9, 10
6.4.1 System (e.g., covert channels, state attacks, emanations)	3, 5, 7, 8, 9, 10
6.4.2 Technology and process integration (e.g., single point of failure, service oriented architecture)	3, 5, 7, 8, 9, 10
6.5 Understand software and system vulnerabilities and threats	1, 3, 5, 7, 8, 9, 10
6.5.1 Web-based (e.g., XML, SAML, OWASP)	3, 5, 7, 8, 9, 10
6.5.2 Client-based (e.g., applets)	5, 7, 8, 9, 10
6.5.3 Server-based (e.g., data flow control)	3, 5, 7, 8, 9, 10
6.5.4 Database security (e.g., inference, aggregation, data mining, warehousing)	5, 7, 8, 9, 10
6.5.5 Distributed systems (e.g., cloud computing, grid computing, peer to peer)	5, 7, 8, 9, 10
6.6 Understand countermeasure principles (e.g., defense in depth)	2, 3, 4, 5, 6, 7, 8, 9, 10

OBJECTIVE	CHAPTER
7.0 OPERATIONS SECURITY	
7.1 Understand security operations concepts	7, 8, 10
7.1.1 Need-to-know/least privilege	1, 2, 10
7.1.2 Separation of duties and responsibilities	1, 2, 9, 10
7.1.3 Monitor special privileges (e.g., operators, administrators)	1, 2, 10
7.1.4 Job rotation	1, 2, 10
7.1.5 Marking, handling, storing and destroying of sensitive information	1, 2, 7, 10
7.1.6 Record retention	1, 2, 10
7.2 Employ resource protection	2, 8, 9, 10
7.2.1 Media management	1, 2, 3, 7, 8, 9, 10
7.2.2 Asset management (e.g., equipment life cycle, software licensing)	1, 2, 5, 7, 8, 9, 10
7.3 Manage incident response	6, 8, 10
7.3.1 Detection	6, 8, 10
7.3.2 Response	6, 8, 10
7.3.3 Reporting	6, 8, 10
7.3.4 Recovery	6, 8, 10
7.3.5 Remediation and review (e.g., root cause analysis)	4, 6, 8, 10
7.4 Implement preventative measures against attacks (e.g., malicious code, zero-day exploit, denial of service)	1, 2, 3, 4, 5, 7, 8, 10
7.5 Implement and support patch and vulnerability management	9, 10
7.6 Understand change and configuration management (e.g., versioning, base lining)	4, 8, 9, 10
7.7 Understand system resilience and fault tolerance requirements	5, 7, 8, 10
8.0 BUSINESS CONTINUITY & DISASTER RECOVERY PLANNING	
8.1 Understand business continuity requirements	1, 4, 6, 8, 10
8.1.1 Develop and document project scope and plan	1, 8
8.2 Conduct business impact analysis	1, 8
8.2.1 Identify and prioritize critical business functions	8
8.2.2 Determine maximum tolerable downtime and other criteria	8
8.2.3 Assess exposure to outages (e.g., local, regional, global)	8
8.2.4 Define recovery objectives	8
8.3 Develop a recovery strategy	8
8.3.1 Implement a backup storage strategy (e.g., offsite storage, electronic vaulting, tape rotation)	4, 7, 8, 10
8.3.2 Recovery site strategies	4, 8, 10
8.4 Understand disaster recovery process	4, 8
8.4.1 Response	4, 8, 10
8.4.2 Personnel	4, 8, 10
8.4.3 Communications	4, 8, 10
8.4.4 Assessment	4, 8
8.4.5 Restoration	8, 10
8.4.6 Provide training	4, 8
8.5 Exercise, assess and maintain the plan (e.g., version control, distribution)	4, 8

OBJECTIVE	CHAPTER
9.0 LEGAL, REGULATIONS, INVESTIGATIONS AND COMPLIANCE	
9.1 Understand legal issues that pertain to information security internationally	1, 6, 8
9.1.1 Computer crime	6
9.1.2 Licensing and intellectual property (e.g., copyright, trademark)	6
9.1.3 Import/Export	6
9.1.4 Trans-border data flow	6, 7
9.1.5 Privacy	6
9.2 Understand professional ethics	1, 6
9.2.1 (ISC) ² Code of Professional Ethics	1, 6
9.2.2 Support organization's code of ethics	1, 6
9.3 Understand and support investigations	6, 8
9.3.1 Policy, roles and responsibilities (e.g., rules of engagement, authorization, scope)	1, 4, 6, 8, 10
9.3.2 Incident handling and response	6, 8, 10
9.3.3 Evidence collection and handling (e.g., chain of custody, interviewing)	6, 8
9.3.4 Reporting and documenting	6, 8, 10
9.4 Understand forensic procedures	6, 8
9.4.1 Media analysis	6, 7
9.4.2 Network analysis	6, 7
9.4.3 Software analysis	6
9.4.4 Hardware/embedded device analysis	5, 6, 7
9.5 Understand compliance requirements and procedures	1, 2, 5, 6, 8
9.5.1 Regulatory environment	1, 4, 5, 6, 8
9.5.2 Audits	1, 5, 6, 8
9.5.3 Reporting	1, 5, 6, 8
9.6 Ensure security in contractual agreements and procurement processes (e.g., cloud computing, outsourcing, vendor governance)	1, 5, 6, 8
10.0 PHYSICAL (ENVIRONMENTAL) SECURITY	
10.1 Understand site and facility design considerations	2, 4, 8, 10
10.2 Support the implementation and operation of perimeter security (e.g., physical access control and monitoring, audit trails/access logs)	1, 2, 4, 8
10.3 Support the implementation and operation of internal security (e.g., escort requirements/visitor control, keys and locks)	2, 4, 8
10.4 Support the implementation and operation of facilities security (e.g., technology convergence)	2, 4, 6, 8, 10
10.4.1 Communications and server rooms	2, 4, 8
10.4.2 Restricted and work area security	2, 4, 6, 8
10.4.3 Data center security	2, 4, 8
10.4.4 Utilities and Heating, Ventilation and Air Conditioning (HVAC) considerations	4, 8
10.4.5 Water issues (e.g., leakage, flooding)	4, 8
10.4.6 Fire prevention, detection and suppression	4, 8
10.5 Support the protection and securing of equipment	2, 4, 8, 10
10.6 Understand personnel privacy and safety (e.g., duress, travel, monitoring)	1, 4, 8

Exam Objectives The exam objectives listed here are current as of this book's publication date. Exam objectives are subject to change at any time without prior notice and at the sole discretion of ISC². Please visit the ISC² Certifications webpage for the most current listing of exam objectives at <https://www.isc2.org/cissp/default.aspx>.

CISSP Training Kit

David R. Miller

Copyright © 2013 by David R. Miller.

All rights reserved. No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher.

ISBN: 978-0-7356-5782-3

Third Printing: June 2014

Printed and bound in the United States of America.

Microsoft Press books are available through booksellers and distributors worldwide. If you need support related to this book, email Microsoft Press Book Support at mspinput@microsoft.com. Please tell us what you think of this book at <http://www.microsoft.com/learning/booksurvey>.

Microsoft and the trademarks listed at <http://www.microsoft.com/about/legal/en/us/IntellectualProperty/Trademarks/EN-US.aspx> are trademarks of the Microsoft group of companies. All other marks are property of their respective owners.

The example companies, organizations, products, domain names, email addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

This book expresses the author's views and opinions. The information contained in this book is provided without any express, statutory, or implied warranties. Neither the authors, Microsoft Corporation, nor its resellers, or distributors will be held liable for any damages caused or alleged to be caused either directly or indirectly by this book.

Acquisitions Editors: Ken Jones and Michael Bolinger

Developmental Editor: Box Twelve Communications

Production Editor: Kristen Brown

Editorial Production: Online Training Solutions, Inc.

Technical Reviewer: Michael Gregg

Copyeditor: Kerin Forsyth

Indexer: Bob Pfahler

Cover Design: Twist Creative • Seattle

Cover Composition: Ellie Volckhausen

Illustrator: Rebecca Demarest

I dedicate this work to Ms. Veronica Leigh Miller and to Mr. Ross Adam Maxwell Miller, sources of enduring warmth, happiness, and pride for me. Forever yours.

Further, I wish to express my deep regret over the loss of Mr. Harold (Hal) F. Tipton, who cofounded (ISC)², the International Information Systems Security Certification Consortium, in 1989. The (ISC)² established and maintains the Certified Information Systems Security Professional (CISSP) certification. Mr. Tipton passed away in March 2012 at the age of 89. This book is also dedicated to him for his vision and leadership in the information technology and IT security industry.

—DAVID R. MILLER

Contents at a glance

	<i>Introduction</i>	xxv
CHAPTER 1	Information security governance and risk management	1
CHAPTER 2	Access control	63
CHAPTER 3	Cryptography	139
CHAPTER 4	Physical (environmental) security	245
CHAPTER 5	Security architecture and design	303
CHAPTER 6	Legal, regulations, investigations, and compliance	365
CHAPTER 7	Telecommunications and network security	415
CHAPTER 8	Business continuity and disaster recovery planning	525
CHAPTER 9	Software development security	577
CHAPTER 10	Operations security	647
APPENDIX A	Additional resources	713
	<i>Index</i>	719
	<i>About the author</i>	771



Contents

Introduction	xxv
Chapter 1 Information security governance and risk management 1	
Where do information security and risk management begin?	2
Security objectives and controls.	5
Understanding risk modeling	8
Understanding countermeasures and controls	10
Reducing the risk of litigation	12
Policies and frameworks	14
Policy documents	15
Risk assessment and management	21
Starting the risk management project	23
Performing the risk assessment	24
Implementing the security program	35
Understanding the new organization chart	36
Understanding the information life cycle	37
Classifying data	38
Implementing hiring practices	45
Implementing termination practices	47
Providing security awareness training	49
Managing third-party service providers	50
Monitoring and auditing	51
Exercises	54

What do you think of this book? We want to hear from you!

Microsoft is interested in hearing your feedback so we can continually improve our books and learning resources for you. To participate in a brief online survey, please visit:

www.microsoft.com/learning/booksurvey/

Chapter summary	55
Chapter review.	55
Answers.	57
Chapter 2 Access control	63
Trusted path.	64
Choices, choices, choices	65
Types of access controls	66
The provisioning life cycle	70
Managing fraud	72
Authentication, authorization, and auditing	74
Identity management	76
Authentication	76
Authorization	103
Auditing	120
Exercises	130
Chapter summary	131
Chapter review.	132
Answers.	134
Chapter 3 Cryptography	139
What is cryptography?	140
The basics of cryptography.	142
Cryptanalysis	143
The strength of a cryptosystem—its work factor	147
Historical review of cryptography	148
Hieroglyphics: 3000 BC	149
The Atbash cipher: 500 BC	149
The Scytale cipher: 400 BC	150
The Caesar or Shift cipher: 100 BC	150
Cryptanalysis: AD 800	151
The Vigenere cipher: AD 1586	152

The Jefferson disk: AD 1795	153
The Vernam cipher/the one-time pad: AD 1917	154
The Enigma machine: AD 1942	154
Hashing algorithms: AD 1953	155
The Data Encryption Algorithm (DEA) and the Data Encryption Standard (DES): AD 1976	156
Diffie-Hellman (or Diffie-Hellman-Merkle): AD 1976	156
RC4: AD 1987	157
Triple DES (3DES): AD 1999	157
The Rijndael algorithm and the Advanced Encryption Standard (AES): AD 2002	157
Other points of interest	158
Cryptographic keys	159
Key creation	160
Key length	160
Key distribution	161
Secure key storage	161
Quantities of keys	162
Key escrow (archival) and recovery	163
Key lifetime or the cryptoperiod	164
Initialization vectors	165
Hashing algorithm/message digest	165
Attacks on hashing algorithms	167
Strong cryptography	168
Symmetric key algorithms and cryptosystems	169
Symmetric keystream ciphers	172
Symmetric key block ciphers	175
Modes of symmetric key block ciphers	180
Signing and sealing using symmetric key algorithms	185
Weaknesses in symmetric key algorithms	189

What do you think of this book? We want to hear from you!

Microsoft is interested in hearing your feedback so we can continually improve our books and learning resources for you. To participate in a brief online survey, please visit:

www.microsoft.com/learning/booksurvey/

Asymmetric key algorithms and cryptosystems	190
Signing by using asymmetric key algorithms in a hybrid cryptosystem	192
Sealing by using asymmetric key algorithms in a hybrid cryptosystem	195
Sending to multiple recipients when sealing	197
Signing and sealing messages	198
Asymmetric key algorithms	201
Cryptography in use	208
Link encryption	209
End-to-end encryption	210
Public key infrastructure	210
Pretty Good Privacy (PGP)	221
Secure channels for LAN-based applications	223
Secure channels for web-based applications	229
Steganography	234
Attacks on cryptography	236
Ciphertext-only attack	236
Known plaintext attack	236
Chosen plaintext attack	237
Chosen ciphertext attack	237
Adaptive attacks	237
Exercises	238
Chapter summary	238
Chapter review	239
Answers	241

Chapter 4 Physical (environmental) security 245

Physical security in a layered defense model	246
Planning the design of a secure facility	246
First line of defense	247
Threats to physical security	247
Liability of physical design	248

Designing a physical security program	249
Crime prevention through environmental design	252
Target hardening	257
Securing portable devices	270
Intrusion detection	272
Heating, ventilation, and air conditioning systems	274
Failure recovery	275
Periodic walkthroughs and inspections	279
Auditing and logging	280
Fire prevention, detection, and suppression	281
Four legs of a fire	281
Fire detection	282
Five classes of fires	283
Sprinkler systems	284
Fire suppression agents	286
Fire extinguishers	288
Fire plan and drill	291
Exercises	292
Chapter summary	293
Chapter review.	294
Answers.	297

Chapter 5 Security architecture and design 303

Identifying architectural boundaries	304
Computer hardware and operating systems	305
Computer hardware	307
The operating system	314
Application architecture	326
Service-oriented architecture	328
Frameworks for security.	332
International Organization for Standardization (ISO)	
27000 series	333
The Zachman Framework for enterprise architecture	334

The Committee of Sponsoring Organizations of the Treadway Commission (COSO)	335
Control Objectives for Information and Related Technology (COBIT)	335
Information Technology Infrastructure Library (ITIL)	336
Generally Accepted Information Security Principles (GAISP)	336
National Institute of Standards and Technology (NIST) Special Publication 800 (SP 800) series	336
Security models	337
Certification and accreditation (C&A)	344
Legal and regulatory compliance	349
Exercises	351
Chapter summary	352
Chapter review	353
Answers	355

Chapter 6 Legal, regulations, investigations, and compliance 365

Computer crimes	366
Is it a crime?	367
A global perspective of laws regarding computer crime	371
The codified law system	371
The common law system	372
The customary law system	373
The difference between laws and regulations	373
Protecting intellectual property	374
Protecting privacy	376
Auditing for compliance	379
Litigation	381
Governance of third parties	382
Software licensing	383
Investigating computer crime	384
When to notify law enforcement	385
Incident response	386
Evidence	396
Forensic investigations	399

Exercises	406
Chapter summary	407
Chapter review	408
Answers	410

Chapter 7 Telecommunications and network security 415

The Open Systems Interconnection (OSI) Model	417
The seven layers of the OSI Model	418
Transmission media and technologies	442
Media types	443
Encoding data into signals	450
Networking topologies	453
Media access methods	459
Network devices	460
Devices within the OSI Model	460
Mainframe computers	463
Client/endpoint systems	464
Remote access by client/endpoint systems	465
Bastion hosts/hardened systems	465
Firewalls	467
Firewalls in use	469
Network address translation	471
Name resolution	473
Dynamic Host Configuration Protocol	474
The virtual private network server	475
Protocols, protocols, and more protocols	475
Internet Protocol version 4	475
Internet Protocol version 6	477
The TCP/IP Protocol suite	478
Commonly used protocols	479
Routing protocols	481
Virtual private network protocols	482
Authentication protocols	484

PAN, LAN, MAN, WAN, and more	485
Personal area networks	485
Local area networks	486
Metropolitan area networks	488
Wide area networks	489
Private Branch Exchange (PBX)	491
Voice over Internet Protocol	491
Wireless networking	492
Attacking the network	505
Types of attacks	505
Exercises	514
Chapter summary	515
Chapter review	517
Answers	520

Chapter 8 Business continuity and disaster recovery planning 525

Disaster recovery plan and the business continuity plan	527
The disaster recovery plan	527
The business continuity plan	528
Stages of the planning process	529
Develop the plans: Proposals	540
Identify preventive controls	541
Develop disaster recovery plans and strategy	541
Developing the BCP (reconstitution guidelines)	560
Presentation to senior management	561
Implementing the approved plans	562
Components of the plans	563
Share the accomplishment with the world?	570
Exercises	570
Chapter summary	571
Chapter review	572
Answers	573

Chapter 9 Software development security 577

- The need for improved security in software 578
- Maturity models 579
 - The software development life cycle 579
 - Project initiation 580
 - Functional design 580
 - System design 580
 - Software development 580
 - Installation and testing 580
 - Operation and maintenance 582
 - Disposal and end of life 585
 - Separation of duties 587
 - Software Capability Maturity Model Integration 587
 - The IDEAL model 588
 - Software development models 588
 - Computer-aided software engineering tools 590
 - Software testing 590
 - Software updating 591
 - Logging requirements 592
 - The software escrow 593
- Programming concepts 595
 - The generations of programming languages 596
 - Object-oriented programming 597
 - Distributed computing 599
- Database systems 605
 - Database models 607
 - Accessing databases 610
 - Polyinstantiation 612
 - Transaction processing 614
 - Increasing the value of data 619
- Attacks on applications 625
 - Lack of validating and filtering data input 625
 - Failure to release memory securely 626
 - Residual maintenance hooks 626

Unintended (covert) communications channels	627
Race conditions	627
Malware	628
Attacking web-based applications	632
Web cache poisoning	634
Hijacking webpages	635
Directory transversal attacks	636
Sensitive data retrieval	636
Malware detection mechanisms	637
Exercises	639
Chapter summary	639
Chapter review.	640
Answers.	642

Chapter 10 Operations security 647

The activities of operations	648
Roles in information technology	649
Remote administration	654
Availability	655
User provisioning	656
Fraud protection	657
Vulnerability assessments	661
Incident response	670
Data management	671
Data classification	671
Media management	672
The media library	672
Maintaining the systems that support the data	673
Data retention	687
Secure deletion	688
Object reuse	689
Secure destruction	690
Fax security	690

Attacks on operations	691
Preventive measures	691
Common attacks and losses	692
Anatomy of a targeted attack	693
Exercises	702
Chapter summary	703
Chapter review	704
Answers	706

Appendix A Additional resources 713

Additional resources available from (ISC)2	713
Miscellaneous additional resources	713
Chapter 1: Information security governance and risk management	714
Chapter 2: Access control	714
Chapter 3: Cryptography	714
Chapter 4: Physical (environmental) security	715
Chapter 5: Security architecture and design	715
Chapter 6: Legal, regulations, investigations and compliance	716
Chapter 7: Telecommunications and network security	717
Chapter 8: Business continuity and disaster recovery planning	717
Chapter 9: Software development security	717
Chapter 10: Operations security	718

<i>Index</i>	719
--------------------	-----

<i>About the author</i>	771
-------------------------------	-----

What do you think of this book? We want to hear from you!

Microsoft is interested in hearing your feedback so we can continually improve our books and learning resources for you. To participate in a brief online survey, please visit:

www.microsoft.com/learning/booksurvey/

Introduction

This is a big book with a lot of pages and not many pictures. You must be serious about achieving this certification. That's good news. This book focuses directly on the security concepts, technologies, and processes that will help you prepare for the most recent edition of the Certified Information Systems Security Professional (CISSP) certification exam. This book presents the wide range of topics you need to know and understand to pass the CISSP exam. As you plow through these chapters, you will recognize how the individual chapters and topics tie together to form a complete and comprehensive framework of security and protection for the valuable information assets of the enterprise. Further, the information you learn from these pages will help you develop the skills and vision necessary to operate as a security professional, helping companies prepare for and defend themselves from the ever-growing array of threats to information systems and business continuity as a whole.

Cybercrime is becoming a more prevalent threat every day. The attacks are becoming more sophisticated and targeted. Financial institutions and governments are being attacked successfully, and the breaches cost companies millions and even billions of dollars in losses. Reports on the breach of Sony's PlayStation Network in April 2011 set the losses at \$3.2 billion USD. In April 2012, more than 1.5 million credit cards were stolen in a breach at a card processing company in Atlanta, Georgia. In 2012, the number of breached customer records containing personally identifiable information (PII) tripled from 2011. Ponemon Institute estimates the average cost of the breach of customers' PII at \$142 per record. In spite of the efforts of diligent network administrators and security professionals, the breaches continue to occur and seem to be much more targeted, sophisticated, and stealthy every day.

There are indications that corporations and even governments are sponsoring the development of attacks on computers and networks, developing new ways to replicate and spread malware, remain hidden, cover the tracks of the malicious presence and activities, and provide remote command and control to compromise any aspect of confidentiality, integrity, or availability at its master's choosing. This new breed of attack is called advanced persistent threat (APT) and is becoming the most challenging and destructive attack ever developed. The defenses of every IT system and network must be carefully considered, well understood, and implemented correctly to provide an appropriate level of protection. Developing this vision and understanding requires the development of skills in a broad range of areas—exactly the focus of the CISSP body of knowledge.

The CISSP certification is becoming an essential piece of the IT and security professional's résumé. Anyone who works in, or hopes to work in, the commercial world or within the government as an IT or security professional will benefit from the knowledge and vision preparing for the CISSP exam will provide. Improving your upward mobility in the organizational

chart, hoping to land that new job, and, in some cases, keeping your existing job might require attaining this certification. For government workers, the Department of Defense (DoD) Directive 8570 has mandated security certifications for everyone at and above a specific pay grade.

NOTE DEPARTMENT OF DEFENSE DIRECTIVE 8570

From the (ISC)² website at <https://www.isc2.org/dod-fact-sheet.aspx>:

This DoD-wide policy, made official in August 2004 and implemented according to the requirements of DoD 8570.1M Manual in December 2005, requires any full- or part-time military service member, contractor, or foreign employee with privileged access to a DoD information system, regardless of job or occupational series, to obtain a commercial information security credential accredited by ANSI or equivalent authorized body under the ANSI/ISO/IEC 17024 Standard. The Directive also requires that those same employees maintain their certified status with a certain number of hours of continuing professional education each year.

This book marches through the 10 domains outlined by the (ISC)²'s Common Body of Knowledge (CBK), providing the topics, concepts, and technologies needed to develop your skills as a security professional and to pass the very challenging CISSP certification exam. Each chapter contains exam tips and practice questions with answers and explanations. In addition, a large collection of questions, answers, and explanations help you become reacquainted with the testing process and identify areas of study requiring more attention.

The book includes the following:

- Information about the prerequisites for the certification
- Guidance on how to prepare for the exam
- Insights on how to analyze exam questions and answer sets on the exam
- A big-picture overview of the CISSP body of knowledge, tying it all together
- Guidance on how to register for the exam
- A 60-minute review of the major topics of the exam to help prepare yourself just before entering the exam center
- A description of the processes you'll witness on the day of the exam
- A description of the additional steps you must perform to become certified after passing the exam

NOTE DOWNLOAD THE REVIEW

Be sure to download the CISSP 60-minute review from <http://aka.ms/cisspTK/examples>.

The author, the editors, and the production team have worked hard to bring you a preparation tool worthy of your time. Take this challenge seriously, commit to the effort, become determined to pass the exam, and achieve certification to propel your career to new heights.

This training kit is designed for information technology (IT) professionals; IT security professional and business managers who support, plan to support, or are responsible for information systems information technologies; and those who plan to take the 2012 edition of the (ISC)² Certified Information Systems Security Professional (CISSP) exam or the SANS Institute Global Information Assurance Certified (GIAC) Information Security Professional (GISP) certification exam.

The material covered in this training kit describes the concepts and technologies targeted on the CISSP exam covering the 10 domains of the (ISC)² CISSP CBK. The topics in this training kit cover what you need to know for the exam as described on the (ISC)² website for the exam. The CISSP exam objectives are available at <https://www.isc2.org/cissp/default.aspx> and in the CISSP Candidate Information Bulletin (CIB) at <https://www.isc2.org/cib/default.aspx>.

By using this training kit, you learn test-worthy concepts and technologies in the following 10 domains of the CBK:

1. Information security governance and risk management
2. Access control
3. Cryptography
4. Physical (environmental) security
5. Security architecture and design
6. Legal, regulations, investigations and compliance
7. Telecommunications and network security
8. Business continuity and disaster recovery planning
9. Software development security
10. Operations security

The CISSP certification is one of the most sought-after requirements in information technology jobs in commercial enterprises and in the government sector. This certification is ideal for mid-level and top-level managers working in or seeking work as senior security engineers, chief information security officers (CISOs) or chief security officers (CSOs). It is assumed that before you begin using this kit, you have met the professional experience requirements for the CISSP certification: a minimum of five years of direct, full-time experience in two or more of the 10 domains of the (ISC)² CISSP CBK; or four years of direct, full-time experience in two or more of the 10 domains of the (ISC)² CISSP CBK plus a four-year degree. In addition, a CISSP certification candidate should have a foundation-level understanding of common business functions, operating systems, applications, and common Internet technologies.

Preparing for the exam

The CISSP exam covers a wide range of concepts and technologies that deal with securing information systems and protecting the enterprise from losses. Because the exam covers such a broad spectrum of concepts and technologies, the best approach to passing the exam is to study sufficiently to understand the concepts and technologies rather than trying to memorize sentences or strings of words.

Most candidates prepare for 6 to 12 months prior to scheduling and taking the exam. Reading a comprehensive book like this one is a key component of preparing for the exam. Poring over practice test questions is another. However, (ISC)² is very focused on protecting the secrecy of the questions in the exam, so don't consider the practice tests, any of them, worthy of memorizing. Nobody but (ISC)² knows the questions and answers on the actual exam. Recognize that the practice test questions are good for two primary targets:

- Getting used to the Q&A process because many of us haven't had to take an exam for years, since school. Studying accustoms you to taking information in. Practice tests accustom you to putting out information, as you have to do on the exam.
- Identifying areas of knowledge that might need additional reading and study if you might not be doing so well with answering the answers correctly.

There are many questions with answers and explanations in this book. Many more are available online, both free and for purchase. When using any free and even inexpensive resources, be a little leery of the declared correct answer, especially if you disagree with it for good cause. If you disagree with an answer to a question, do your homework and verify your understanding and don't cave in just because someone else, who may know substantially less than you, wrote and uploaded some misguided question and answer. If you can justify your answer, stick to your understanding and your answer. This is the sign of competence: trusting in your knowledge.

As indicated before, many candidates spend 6 to 12 months preparing by reading books and related documentation and reviewing hundreds of questions, answers, and explanations and then, within a month of actually sitting the exam, they attend a 5-day or 6-day CISSP class that is more like a boot camp. Attending a class like this is putting the polish on the apple. It brings a fresh perspective and provides a grasp of the big picture, pulling it all together for you just at the right time. Attending a class like this also provides a live person, usually highly skilled, rehearsed in the CISSP body of knowledge, and with security vision, to answer questions to clarify your understanding of some of those fuzzy areas. Try to take the exam relatively soon after sitting such a class. The information you need to pass the exam tends to diffuse over time as you become consumed once again by your real-world, daily activities.

Many students make the mistake of answering the questions based on their specific daily activities and procedures. Remember that these practice tests and this certification exam have been developed to encompass every type of enterprise, organization, business, and agency in the commercial world as well as in government and military IT environments. Many questions target the more generic, complete, and comprehensive answer that would apply to every type of computing environment. Because many exam questions include multiple potentially correct answers, don't add the phrase "on my job" to the question. The security structure and practices used daily in your specific environment are only one example or implementation of many types of appropriate security solutions and implementations. The larger, superset type answer that would apply to any IT environment is often the correct answer. On the exam, consider an international solution or recommendation first instead of a country-specific recommendation. Then verify that the international solution satisfies the nature of the issue in the question best. Consider the answer "Use a firewall" for improved LAN security, instead of "Use (some specific) firewall rule" unless the question is targeting the specific rule.

As the day of the exam draws near, focus more on the practice questions and less on trying to read new material. At this point, you probably know as much as you can cram into your head on this topic. The questions will accustom you to putting out information instead of absorbing information. At least one week before your exam, review Appendix B, "CISSP 60-minute review," (available at <http://aka.ms/cisspTK/examples>), and be familiar with every term on the list. The week before your exam, try to work through 250 to 400 questions in a single session to develop a sense of the endurance required on exam day.

Signing up for the exam

Details about registering for the CISSP certification exam can be found on the (ISC)² website at <https://www.isc2.org/certification-register-now.aspx>. A list of where and when the exams are provided is available on this site. Filter this list of scheduled exams by the exam type (CISSP), the state (optional), and country (optional). When the list appears, verify that the one(s) you are interested in are not sold out, private, or closed. As of June 1, 2012, (ISC)² exams are

offered as computer-based exams through Pearson VUE testing centers, and the historic paper-based exams are rarely offered. You should be able to schedule the exam one day and take it the next day through Pearson VUE.

At some point, you'll need to create a login with (ISC)², agree to its terms and conditions, and complete the registration form. You must receive approval from (ISC)² before you are allowed to schedule a computer-based exam. The most recently published fees are as follows:

- For the Americas, Asia/Pacific, Africa, and Middle East nations: \$599 USD
- For European nations: €520 EUR
- For the United Kingdom: £415 GBP

(ISC)²'s pricing can be found at https://www.isc2.org/uploadedFiles/Certification_Programs/exam_pricing.pdf.

If you qualify for special arrangements such as for disabilities, languages, or the necessary use of a translation dictionary, contact (ISC)² directly at Customer Support at +1-866-331-4722 (toll free in North America) or +1-727-785-0189 (outside North America). Requests may also be made by email to registration@isc2.org or by fax to +1-727-683-0785. Some details can be found at <https://www.isc2.org/cancel-policy.aspx>.

You should review and understand the somewhat strict cancellation/rescheduling/retake policy, which you can find at <https://www.isc2.org/cancel-policy.aspx>.

The exam itself

This will be one of the most difficult exams you have ever taken. The following list provides an overview of the exam, and Table 1 explains the retake policy.

- **Number of questions** 250
- **Type of question** Multiple choice
- **Number of answer choices** 4 potential answers
- **Number of correct answers per question** 1 correct answer
- **Number of questions being graded** 225
- **Points per question** Weighted, based on difficulty
- **Allotted time** 6 hours
- **Questions provided by** Computer-based exam

TABLE 1 CISSP exam retake policy

Exam attempt	# of days before a candidate can retake the exam
Second	30 days
Third	90 days
Fourth	180 days

There are several versions of the exam so that no two adjacent candidates will have the same questions. The exam consists of 250 multiple-choice questions with 4 possible answers but only one correct one. Many questions are looking for the best answer of two or more answers that could be considered correct. Some questions seem to be looking for the best answer of four incorrect answers. In a few places on the exam, multiple questions (two or three) are based on a simple scenario described in the first question of the series. Many of these “best of the multiple correct or no correct answers” are based on the concepts of the superset and its subsets. For example, consider the following question:

- 1.** As part of the security program, you must mitigate the likelihood and impact of a security breach. Which of the following should you perform?
 - A.** A physical inspection of the facility
 - B.** Risk assessment
 - C.** Risk management
 - D.** A review of the security policies of the organization

In this case, all four answers are good answers, each correct on its own. However, your challenge is to pick the one best answer. To do this, recognize that risk management includes a risk assessment. Risk management begins with the assessment so you understand where the risks are, but it also includes the development of proposed countermeasures and the implementation of approved countermeasures, when you are actually managing the risks. The risk assessment will likely require a physical inspection of the facility and a review of the organization’s security policies. Risk management is the superset, and the risk assessment is a subset. Risk management is the bigger, more correct answer. As you are preparing for the exam, remember to identify these superset–subset relationships. During the exam, recognize these relationships in the answers and then verify the specific target of the question. The question will often be focusing on the larger, more comprehensive superset answer, but reconfirm this by rereading the question after you establish the superset–subset relationships in the answer set.

Only 225 questions of the 250 questions are counted against your grade, and these are weighted based on their level of difficulty. The other 25 questions are being qualified (or disqualified) as viable questions to be used in future exams. The candidate has six hours to complete all questions, so it is a long day in the exam center. Don’t forget to breathe. Every

now and again, sit back, relax, stretch, take another deep breath, and then hit the exam again, refreshed and confident in your knowledge.

Proceed through the exam one question after the other, in order. If taking the computer-based exam, (ISC)² currently reports that you can go back to earlier questions one at a time. Upon completion of the exam, after all 250 questions have been answered, you can jump directly to any question by number.

You will be required to perform one or more mathematical calculations. A calculator is provided on the computer-based exam. The calculations will be addition, subtraction, multiplication, and division, such as those you'll see when calculating the single loss expectancy (SLE) or the annualized loss expectancy (ALE) for some potential incident.

That sounds ugly enough already, but it is time now for the really ugly part. The questions and answers are filled with subtleties and ambiguity, making this exam one that requires excellent reading and comprehension skills. Some questions aim at solid, reasonably obvious or direct targets, but many questions aim at ancillary targets, not necessarily the key concepts of the technologies. Many questions are vague or difficult to understand what the question is really asking, and many answers mirror that ambiguity.

Overall, the exam seems designed not only to gauge the candidate's knowledge of the topic but also to confuse and demoralize him, degrading his confidence, as if trying to convince him that he has already suffered a defeat. After teaching the CISSP courses for about 10 years, I don't recall ever hearing a student describe feeling good about the exam after taking it. It seems no one is ever comfortable that she has passed; most feel that they probably have failed. This seems to be the desired outcome—wear the candidate down and convince him to concede. "Stop the pain. Just quit. You can always retake the test in six months or a year."

Although this makes the CISSP exam ugly, difficult at best, it is the norm. Get used to the idea that you will not enjoy taking this exam and that you must rise to the challenge and overcome. Remember that the penalty for conceding and giving up during the exam is that you will probably have to (or want to) retake the exam later on. Let the thought of having to retake the exam drive you onward aggressively, with determination to complete it with the best you have to put out. During the exam, when you bump into a question whose correct answer completely eludes you, pat yourself on the back and consider that you just ran into another one of the 25 questions that will not be graded. Do your best to answer these questions correctly but do not beat yourself up for having to guess at an answer. Have confidence in your knowledge. You have studied hard, and you know your stuff. March forward through all 250 questions with confidence and unflinching determination.

The day before the exam, try to spend several hours working practice test questions and reviewing this study kit along with your notes and additional resources, especially in areas that still might not be clear. Nevertheless, allocate your time so you have one or two hours to relax and get away from the work. Then get to bed early and get a good night's rest. You'll

need to be up early on exam day. Set and recheck your alarm clock. Most paper-based exams begin candidate check-in at 8 A.M., with testing beginning at 9 A.M. You'll want to get there early to reduce the stress of running late to an important appointment and to provide time, approximately one hour, for a final review (included in Appendix B, which you can download at <http://aka.ms/cisspTK/examples> or find on this book's practice CD). Consider the time necessary for getting ready, breakfast, traffic, roads, traveling distance, parking issues, and the hour for the review before the exam check-in time. With that list in mind, set your alarm clock appropriately.

Consider whether you want or need to take a snack with you to the exam center to survive the six-hour exam time.

Seeing the big picture of CISSP

The security structure of the organization should be designed to protect the *confidentiality*, *integrity*, and *availability* (CIA) of the organization's valuable information assets effectively. These are three distinct security objectives and often require different types of countermeasures for their protection. Some information assets need to be kept secret (confidentiality) like the recipe for grandma's secret sauce. Other information assets are public information but must remain highly accurate (integrity) like the trading prices of stocks. In most cases, the information must also remain accessible (availability) to business managers and workers when they need it so they can make the best business decisions to optimize the profits of the organization.

Many of the concepts presented within the CISSP CBK and tested on relate closely to prudent business management—aspects of running a business such as establishing a security program, defining the security program through policies, performing risk management, planning disaster recovery and business continuity, and managing compliance with relevant laws and industry regulations. Business management must perform due diligence and due care prudently to avoid being negligent and liable for preventable losses. Anonymous CISSP exam candidates have reported a high concentration of exam questions targeting the disaster recovery planning (DRP) and business continuity planning (BCP) processes. These concepts fall largely into the category of *administrative controls*.

Much of the exam is technology-centric, such as secure computer hardware design—secure operating system design, secure application development, networking technologies and architecture, and cryptography. These fall into the category of *technical* (also called *logical*) *controls*.

Still, a large piece of the exam focuses on physical security such as the location, design, and construction of the facility, security guards, physical access controls, and fire protection. These fall into the category of *physical controls*.

In addition to the recognition of superset–subset relationships, several key concepts permeate the bulk of the CBK. Keep these concepts in mind as you dissect and analyze the way-too-many questions on the exam.

- It might not be presented everywhere it is relevant, but remember that *human safety is always the top priority*.
- The most ethical answer will likely be the correct answer unless it causes injury or risk to people's safety.
- Senior management must drive the security program by requiring its development and assigning the appropriate level of responsibility, authority, and support to the security team.
- Although many topics include security in the government setting, the CISSP is largely focused on the commercial business. The primary goal is to *maximize the profits of the business* by avoiding losses (preventive controls), reducing losses when a breach does occur (disaster recovery), and never letting the company fail by going out of business due to some disaster (business continuity).
- Every security control must be cost justified, weighing the cost of the control against the financial benefits that the control will provide. Accurately performing this cost justification for every proposed control helps ensure satisfaction of the requirement to maximize profits.
- All decisions are made by management. Security professionals simply provide quality input, vision, understanding, options, and cost versus benefit analyses and then request approval from senior management before implementation or action.
- Implement every aspect of security by following the principle of least privilege, assigning just the barest minimum of privilege required for the user to perform her authorized duties. Review the assigned privileges for all users to avoid security-related conflicts of interest and defend against insider fraud.
- No single countermeasure or control will provide complete and adequate security. Multiple layers of security are required in every solution.
- The use of automated tools helps deal with the complexity of planning, assessments, audits, and recovery.
- Department managers must be the enforcers of the security policies. They operate in every facet of the organization. They know the people, the processes, and the detailed activities occurring within their area of responsibility in the organization on a daily basis.
- Users tend to be the weakest link in the security of ongoing daily operations. To mitigate this vulnerability, security awareness training is essential for every user. Users with increased access or privilege require specialized and more frequent security awareness training.

- The first tier of management, the tier closest to the workers, is responsible for the dogmatic and consistent monitoring and enforcement of policy. Failure to remain aware and enforce policy consistently is a recipe for negligence and discrimination lawsuits.
- Senior management must perform due diligence and due care and manage the enterprise prudently to mitigate risks and avoid being negligent, a critical component of litigation against the enterprise.

The information in the CBK is divided into 10 domains, but these domains have close relationships to one another. When you understand how these domains relate, the big picture becomes clearer. There might be some overlap in some of the topics, which are covered in more than one domain. This demonstrates that these 10 topics are related. Don't let the redundancy bother you. Gain a second perspective on the topic as it relates to the nature of the other domains. The order of the domains is also not something to be concerned about. This book covers all known areas of knowledge the CISSP certification exam requires and (hopefully) presents these concepts and technologies in a manner that allows a more linear and logical flow and progression of the information.

Domain 1, on information security governance and risk management, introduces the security program in the organization. The security program is defined by policies and other documentation. These policy documents establish the security posture of the organization and should include the specific and sometimes unique security concerns but must also include all the laws and regulations that are applicable to the organization. These laws and regulations are covered in Domain 6 on legal, regulations, investigations and compliance, so recognize that relationship. Domain 1 continues with risk management and the risk assessment. Risk assessment, with the inventory of information assets it is based on and the valuation of those information assets, will be useful later, in Domain 8 on business continuity and disaster recovery planning (BCP and DRP). This is the business or administrative side of CISSP.

Domain 2, on access control, introduces the three main control categories: administrative controls, technical controls, and physical controls.

Administrative controls define the policies concerning how workers are supposed to behave and make decisions. The rules are documented but are also well known by all users through security awareness training and consistently enforced by management.

Technical controls begin with the fundamental understanding that, today, our valuable information assets are stored on, transmitted between, and processed on computers and computer networks. Therefore, users must use computers, operating systems, and applications to access those assets. If we hope to implement prudent security controls to protect those assets, we must establish that the computer, its operating system, and its applications will not cause security violations. This level of trust in the technological equipment is addressed in Domain 5, on security architecture and design, and in Domain 9, on software development security, so that additional technical controls that can be implemented, such as cryptography

(Domain 3), networking technologies (Domain 7), and system fault tolerance and redundancy (Domain 10) in operations security can be effective.

The physical aspect of securing valuable information assets requires the facility to remain secure, reasonably well protected from natural threats such as tornados and fire; from human threats such as burglars, an internal thief, or a social engineer; and from failures in the supply system that keeps the physical environment safe and functional. Domains 4 and 10 address these types of physical threats and the related controls.

The day of the exam

The day of the exam has arrived. Be sure you don't oversleep. Wake up early enough to provide time for the following tasks, which you should perform before checking in for your exam.

- Get up, get ready, and be awake.
- Have something to eat. This will be a long day, and there isn't much of an opportunity for a real break until after you complete the exam, possibly six hours after entering the exam center. Put some food in your stomach and, if you're into caffeine, you'll want a healthy supply in your belly.
- Get your materials together before leaving the house. This includes a photo ID; a printout of your exam registration letter; address and directions to the testing center, including floor number and room number; your study notes; and this book for your final review. Bring your snack if you decided you want one.
- Figure out a secure place to keep things you might normally carry with you that won't be allowed in the testing center such as your purse, laptop, cell phone, suit case, pager, pens, pencils, iPod, headphones, and so on. It is kind of like going through security at the airport. Nothing in your pockets.
- Get to the testing center one and a half hours before the candidate check-in time. Use this time to review Appendix B, "CISSP 60-minute review." Spend approximately one minute per page reviewing the terms and recalling as many details as possible for each item on the list. Review any other notes you consider important to you.
- Leave all but your ID, exam registration letter, and snack someplace secure. Enter the testing center and get to the exam check-in location. Identify the location of the restrooms (and perhaps take a final pause because you won't want to step out of the potentially long check-in line later).
- Check in. This is typically a very rigid and formalized process, the continuation of the induced stress and demoralization that seems to permeate the CISSP testing process. After the check-in personnel carefully check your ID and your exam registration letter, they'll give you the once over, looking for contraband, anything that isn't required. Any identified items must be dispensed with, one way or another. Finally, they'll walk you

into the testing room and place you in your preassigned seat. You must sit here quietly, patiently waiting until all have been checked in, inspected, escorted, and quietly seated.

- For paper-based testing, at or near the scheduled testing time, the proctoring team will pass out the various versions of the sealed exam booklet, the Scantron answer sheet, and one or two #2 pencils. The lead proctor (a local CISSP volunteer), will begin to explain how the exam is conducted—more formality and induced stress. Instructions will be provided on filling out the Scantron sheet with codes, names, dates, and so on. Paper-based exams are rarely issued but might still be used when computer-based testing is not an option.
- As soon as all the formalities (and stress) have been dumped on the candidates, the lead proctor will announce that you may break the seal on your copy of the exam booklet or begin the computer-based exam. The test has just begun. Take your time. The exam is not a race but keep your eye on the clock to be sure your pace allows you to answer all 250 questions in the six-hour timeslot. You must average 42 or more questions per hour to complete the exam in the allotted time.
- Try to avoid adding “in my job” to every question. Candidates preparing for this exam are often in this IT line of work already and are familiar with many of the concepts and technologies presented. The exam is not about your specific job but about general security concepts that apply to virtually every business. Some candidates must learn to dumb down a little in their areas of expertise and go with the more generic CISSP answer on the exam. This exam targets every business, not the one you are currently thinking of.
- Identify the discriminators that lead you to the one best answer and the distractors that deliberately add confusion to the question and might lead you to the wrong answer. References to specific industry niches, products, or vendors are usually distractors of this type. Evaluate whether these details have any relevance to the answers and ignore them when they don't.
- In some cases, there are multiple correct answers. Identify broader superset answers and the more specific subset answers. Many times, the test is looking for the best answer, which will probably be the broader, more comprehensive correct answer—the superset answer. After you identify the superset answer, reconfirm that it still matches all the discriminators in the question and then go with it. An example might be a question about designing secure communications between the headquarters and a branch office. Answers might include a VPN, an IPsec tunnel, an L2TP tunnel, and an SSH tunnel using public key/private key authentication. The VPN includes these other more specific VPN types, so it is the broader, more comprehensive, superset answer. Next, verify that there isn't anything in the question to cause the VPN superset answer to be incorrect, such as a requirement for symmetric keys and security associations. (Therefore, IPsec would be more correct.)

- In some cases, you might not find an answer you feel is correct in the four choices. Review the question and substitute similar words or phrases that might be more commonly used in the IT environment to make one of the answers correct. After exercising this technique for a minute, assume the most likely substitution is accurate and choose the correct answer.
- Remember to breathe. Every now and again, sit back and take a deep breath. It will help keep your head clear and minimize fatigue.
- Bathroom breaks are allowed, but typically, only one candidate may go at a time to avoid the potential for consorting and cheating. You must raise your hand, be recognized, carefully collect your testing materials, sign out, be escorted to the restroom and back, sign in, be escorted to your preassigned seat, and then allowed to resume your exam. The clock does not stop. The time you take on bathroom breaks is included in the six-hour time limit, so be brief about it.
- (ISC)² does allow the quiet consumption of snacks during the exam, but you must bring the items with you, leave the items in the back of the testing room, raise your hand and be escorted to your snacks, quietly consume the snack, and then be escorted back to your exam. The time you take on food is included in the six-hour time limit, so, again, be brief about it.

After completing the exam

When taking the computer-based exam, the results are typically available immediately after completion of the exam, or you might be required to see the testing center representative for your results. If your effort is successful, you will be informed that you passed the exam. Congratulations! Typically, passing scores are not provided. If you failed the exam, your score, and a breakdown by domain is provided. This helps guide your studies as you prepare for the second certification attempt. Paper-based exam results are provided to candidates by email. This process can take up to six weeks but often takes one or two weeks.

Upon successfully completing and passing the (ISC)² CISSP exam, the CISSP candidate must complete the certification requirements by answering several questions regarding criminal history and related background and complete the (ISC)² CISSP Candidate Agreement, describe the related experience, attest to the truthfulness of the affidavit, and commit to adhering to the (ISC)² Code of Ethics. Further, the (ISC)² CISSP candidate must acquire a signed endorsement from a holder of CISSP certification, attesting to the accuracy of the candidate's assertions regarding professional experience and good standing within the community to the best of that person's knowledge. (ISC)² randomly performs audits of about 10 percent of candidates on the claimed professional experience and criminal history.

NOTE CONTINUING PROFESSIONAL EDUCATION CREDITS

Upon certification, the CISSP must acquire and record 120 continuing professional education (CPE) points every three years, as outlined on the (ISC)² website at <https://www.isc2.org/maintaining-your-credential.aspx>.

CISSPs must not have violated the (ISC)² Code of Ethics and must submit the annual maintenance fee of \$85 USD.

Using the companion CD

A companion CD is included with this training kit and contains the following:

- **Practice tests** You can reinforce your understanding of the topics covered in this training kit by using electronic practice tests that you customize to meet your needs. You can practice for the CISSP certification exam by using tests created from a pool of 250 practice exam questions, which give you many practice exams to ensure that you are prepared.
- **Appendix B** An electronic version of the 60-Minute Review is included for you to study before you take the CISSP certification exam.

How to use the practice tests

To start the practice test software, follow these steps.

To install the practice test software for a training kit, open Program And Features in Control Panel.

1. Click Start, choose All Programs, and then select Microsoft Press Training Kit Exam Prep. A window appears that shows all the Microsoft Press training kit exam prep suites installed on your computer.
2. Double-click the practice test you want to use.

When you start a practice test, you choose whether to take the test in Certification Mode, Study Mode, or Custom Mode:

- **Certification Mode** Closely resembles the experience of taking a certification exam. The test has a set number of questions. It is timed, and you cannot pause and restart the timer.
- **Study Mode** Creates an untimed test during which you can review the correct answers and the explanations after you answer each question.
- **Custom Mode** Gives you full control over the test options so that you can customize them as you like.

In all modes, the user interface when you are taking the test is about the same but with different options enabled or disabled, depending on the mode.

When you review your answer to an individual practice test question, a “References” section is provided that lists where in the training kit you can find the information that relates to that question and provides links to other sources of information. After you click Test Results to score your entire practice test, you can click the Learning Plan tab to see a list of references for every objective.

How to uninstall the practice tests

To uninstall the practice test software for a training kit, open Program And Features in Control Panel.

Acknowledgments

The author’s name appears on the cover of a book, but I am only one member of a much larger team. Each of my managers, editors, and reviewers contributed significantly to this book. I wish to thank them each, and I hope to work with them all in the future.

- **Michael Bolinger** O’Reilly Media, Inc.
- **Dan Fauxsmith** O’Reilly Media, Inc.
- **Kara Ebrahim** O’Reilly Media, Inc.
- **Kristen Brown** O’Reilly Media, Inc.
- **Jeff Riley** Production, Box Twelve Communications
- **Michael Gregg** Technical reviewer, Superior Solutions, Inc.
- **Kerin Forsyth** Copy editor, Online Training Solutions, Inc. (OTSI)
- **Jaime Odell** Project Manager, OTSI
- **Neil Salkind** Literary Agent, Studio B

I’d also like to thank the following individuals for their initial work on the project:

- Ken Jones
- Carol Vu
- Susan McClung
- Kurt Meyer

I also wish to thank Ms. Shon Harris, a mentor, a professional associate, and a dear friend. You’re the best, Shon.

Support and feedback

The following sections provide information on errata, book support, feedback, and contact information.

Errata

We've made every effort to ensure the accuracy of this book and its companion content. Any errors that have been reported since this book was published are listed at:

<http://aka.ms/cisspTK/errata>

If you find an error that is not already listed, you can report it to us through the same page.

If you need additional support, email Microsoft Press Book Support at:

mspinput@microsoft.com

Please note that product support for Microsoft software is not offered through the preceding addresses.

We want to hear from you

At Microsoft Press, your satisfaction is our top priority and your feedback our most valuable asset. Please tell us what you think of this book at:

<http://www.microsoft.com/learning/booksurvey>

The survey is short, and we read every one of your comments and ideas. Thanks in advance for your input!

Stay in touch

Let us keep the conversation going! We are on Twitter at *<http://twitter.com/MicrosoftPress>*.

Cryptography

For thousands of years, humans have needed to keep secrets, whether the secret had to do with the strategy for the attack on a walled city, conspiracies to overthrow some unpopular political system, or the protection of credit card numbers that are transmitted during online purchases. So for these thousands of years, we have continued to design and use cryptosystems to help keep these valuable secrets secret.

Exam objectives in this chapter:

- 5.1 Understand the application and use of cryptography
 - 5.1.1 Data at rest (e.g., Hard Drive)
 - 5.1.2 Data in transit (e.g., On the wire)
- 5.2 Understand the cryptographic life cycle (e.g., cryptographic limitations, algorithm/protocol governance)
- 5.3 Understand encryption concepts
 - 5.3.1 Foundational concepts
 - 5.3.2 Symmetric cryptography
 - 5.3.3 Asymmetric cryptography
 - 5.3.4 Hybrid cryptography
 - 5.3.5 Message digests
 - 5.3.6 Hashing
- 5.4 Understand key management processes
 - 5.4.1 Creation/distribution
 - 5.4.2 Storage/destruction
 - 5.4.3 Recovery
 - 5.4.4 Key escrow
- 5.5 Understand digital signatures
- 5.6 Understand non-repudiation

- 5.7 Understand methods of cryptanalytic attacks
 - 5.7.1 Chosen plain-text
 - 5.7.2 Social engineering for key discovery
 - 5.7.3 Brute Force (e.g., rainbow tables, specialized/scalable architecture)
 - 5.7.4 Cipher-text only
 - 5.7.5 Known plaintext
 - 5.7.6 Frequency analysis
 - 5.7.7 Chosen cipher-text
 - 5.7.8 Implementation attacks
- 5.8 Use cryptography to maintain network security
- 5.9 Use cryptography to maintain application security
- 5.10 Understand Public Key Infrastructure (PKI)
- 5.11 Understand certificate related issues
- 5.12 Understand information hiding alternatives (e.g., steganography, watermarking)

What is cryptography?

Cryptography is defined as hiding the meaning of a message and the revelation of that meaning at some later time or other place. As security needs became more complex and cryptographic algorithms evolved into sophisticated cryptosystems, cryptography grew to provide more than just confidentiality services. Contemporary cryptosystems provide five major services:


- **Confidentiality** Encryption provides confidentiality: *keeping the secrets secret*. Confidentiality can provide a form of access control or authorization. If a message is encrypted and the decryption key is only provided to those who are authorized to read the message, then those without the decryption cannot access the information within the message.
- **Authentication** Cryptography can provide a *claim of identity* by a sender and a *verification of that claim* by a recipient to authenticate the sender. Symmetric key cryptography provides a weak form of authentication, and asymmetric key cryptography can provide a strong form of authentication.
- **Nonrepudiation** If the identity of a sender can be strongly verified (proven) through strong authentication techniques, then not only does the recipient know the sender's identity, but *the sender cannot deny being the sender*. As stated previously, asymmetric key cryptography can provide this authentication mechanism, identifying the sender strongly enough to provide nonrepudiation. Symmetric key cryptography cannot provide nonrepudiation because it provides a weak form of authentication, as described in the "Symmetric key algorithms and cryptosystems" section later in this chapter.

- **Integrity** When the integrity of information is known to be good, the information is believed to be *complete, consistent, and accurate*. Information considered to have high integrity is information that is considered not to have been altered or tampered with and that can be trusted to some extent. As a security objective, integrity has two components: *integrity protection* and *integrity verification*.


Cryptography (confidentiality) can protect the integrity of a message by encrypting the message. If an unauthorized person cannot access and read or understand the meaning of a message, that person cannot intelligently modify the meaning of the message, so the integrity of the message has been protected.

Cryptography provides integrity verification by calculating a message digest (also called a hash value) of the message at the time of its creation, when the integrity of the message is known to be good. This message digest acts as a fingerprint of the message. Then, at the time of use, a new message digest is calculated. If these two digest values are identical, this proves that the message has not been altered since the time of the creation of the message. The integrity of the message has been verified.

- **Secure key distribution** Symmetric key cryptography requires both the sender of a message and the recipient of the message to have a copy of the key used to encrypt the message. Somehow, the sender must provide this copy of the encryption key to the recipient, and this must be accomplished in a secure manner. Contemporary cryptosystems can provide this secure key distribution service.



Sometimes the objective is to protect sensitive (valuable) content from others while you store it for use at a later time (*data at rest*). Examples might include the protection of the secret sauce recipe for your top-selling product or protection of your collection of passwords in your password vault. Whole disk encryption and smaller encrypted volumes might be used to provide protection for data at rest. An example of a whole disk encryption tool is BitLocker Drive Encryption, a Microsoft product. An example of an encryption tool for whole disk and for smaller encrypted volumes is the open source TrueCrypt application.



Other times, you must communicate your valuable information over hazardous network connections or other communications channels, such as by broadcasting radio signals into the air, and you are concerned that someone might steal the information. This is *data in transit*. A virtual private network (VPN), such as the Microsoft Point-to-Point Tunneling Protocol (PPTP) or the open standard Internet Protocol Security (IPsec), provides protection for data in transit from local area network (LAN) to LAN. Secure Sockets Layer (SSL) provides protection for data in transit over the Internet between anonymous clients and servers.

Cryptography can help solve both of these types of problems. The developers of applications, operating systems, and networking services and protocols cryptographically enable their software by including cryptographic code and linking to cryptographic application programming interfaces (APIs) that might be available.



There is another area of protection of data. As the user is actively accessing and using the information, sensitive data can be presented on video displays or in printed reports. This is an example of *data in use*. Data in this state cannot be protected by cryptography because it must be in a human-readable form. It must be protected by physical security measures and the user's awareness and defenses against social engineering, described in the "Cryptanalysis" section later in this chapter.

In this chapter, you explore the history of cryptography and the evolution of cryptosystems over time. You review symmetric key algorithms, asymmetric key algorithms, and hashing algorithms, and then you study the combination of these technologies in contemporary, hybrid cryptosystems that can provide all the required protective services. Cryptographic functions are added to the discussion as necessary to provide a balance between the need for strong security and the price paid in processing time to perform the various cryptographic functions.

The basics of cryptography



Cryptology is the study of all things crypto. This includes *cryptography*, the process of hiding the meaning of a message and revealing it at a later time, as well as cryptanalysis, cryptographic algorithms, cryptosystems, and key management, for example.

Although contemporary cryptosystems use many components and processes, strictly speaking, cryptography requires three components to operate:



- **A cipher (also called an algorithm)** A data manipulation process
- **A key (also called a cryptovariable)** Kept secret and used to impose a unique randomness on the message
- **A plaintext message** In need of protection



Encryption converts the plaintext message into *ciphertext*. The plaintext message is processed by the cipher, and the processing is altered by the encryption key. Ciphertext is (generally) not readable by humans; therefore, it provides secure storage and transmission over otherwise untrusted communications channels. The message is recovered (made readable by humans again) by *decrypting* the ciphertext into the readable message, this time by using the decryption key, as shown in Figure 3-1.

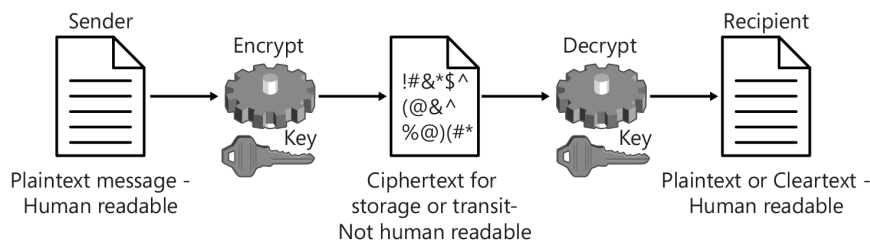


FIGURE 3-1 The encryption and decryption processes

The cryptographic algorithm (cipher) is the series of manipulations (logical or mathematical steps) performed on data to encrypt and decrypt the content. You can consider the cryptographic algorithm (cipher) to be like the engine in a car. Consider the cryptosystem to be the car that is built around that engine to provide the complete collection of functions that might be desired to satisfy the security objectives. The cryptosystem will include a cryptographic algorithm and might include many other components, such as symmetric algorithms, asymmetric algorithms, hashing algorithms, one or more key pairs, and other randomizer-type variables, such as an initialization vector, a salt, a seed, or a nonce.

The car (cryptosystem) needs a well-built engine (cipher), but the engine alone does not provide all the services needed, so additional components are added to the strong engine to increase the capability of the car and improve its utility and performance.

The goal is to build or implement a cryptosystem that provides as many of the five desirable cryptographic functions (confidentiality, authentication, nonrepudiation, integrity, and secure key distribution) as needed and to do it in a way that *adequately protects* the data without spending more time and money than is required to process the data through the cryptosystem.

Tying this all together, as described in Chapter 1, “Information security governance and risk management,” all valuable information assets were identified, and the value of each information asset was determined during the risk assessment. Then, following policy, a specific level of protection was required for each data classification level based on the asset’s value. The cryptosystem must be designed to provide the cryptographic portions of that protection at the level of protection required by the policy of the organization. Therefore, policy defines what the term *adequate protection* means. Providing stronger protection costs more in technology (cryptographic hardware and software) and in time to perform the process through the cryptosystem. Implementation of the right level of protection, balanced against the additional costs that go along with stronger protection, should be used to implement the correct cryptosystem.

Generally, as functions are added to a cryptosystem to improve its strength, the performance of the system degrades and the cost of using the system increases. All security measures, including cryptography, must be cost justified. Implement strong enough cryptography to satisfy the security needs for the data being protected but without spending more on the cryptography than can be justified.

Cryptanalysis



Cryptanalysis, a subset of cryptology, is the science or process of cracking a cryptosystem. Cryptanalysis is performed by the good guys and by the bad guys. The bad guys want to steal confidential messages to gain unauthorized access to valuable information or, better yet, to crack or reveal decryption keys so they can steal many confidential messages.



The good guys try to crack cryptosystems and reveal messages and keys to identify and validate the strength of the cryptosystem. They measure the amount of time and resources necessary to reveal messages or keys. The time and resources required to crack a cryptosystem is defined as the *work factor* of the cryptosystem. The larger the work factor, the stronger the cryptosystem.

One type of attack to crack a cryptosystem is the *brute force attack*, in which every possible key is tested to see whether it accurately decrypts the message. This is an exhaustive attack and typically takes a very long time, but it is guaranteed to be successful eventually.

A smaller subset of the brute force attack is the *dictionary attack*, by which the attacker uses a list of words from a dictionary to test as the password. Because many users choose passwords or keys that are easy to remember, such as commonly used words found in a dictionary, this attack is often successful in a relatively short period. These words are tested as the password until a match is identified or the words on the list are exhausted.

A combination of the brute force attack and the dictionary attack is the *hybrid attack*. In this attack, words from the dictionary are tested as the password or key. If a word fails, a collection of additions and substitutions is made to the word to produce variations, such as:

- Mustang
- Mu5tang
- mu\$tang
- Must/\ng
- &&Mu\$tang!!

These variations are then tested as the password until a match is identified or until the word list and the set of variations are exhausted.

A relatively new attack on passwords is the *rainbow attack*. Many applications and operating systems that include a key store or user database avoid storing plaintext keys and passwords by calculating and storing the message digest of the password. Message digests, also known as hash values, are covered in the “Hashing algorithm/message digest” section later in the chapter. Message digests are like fingerprints of the message (key or password in this case) and cannot be directly reversed to reveal the original input value (the plaintext password). These message digests are also present wherever cached keys or user credentials might be stored on a computer system.

To perform the rainbow attack, attackers must first perform a brute force attack that includes every combination of characters for a specific length of password, and they record the message digest of each combination. This table of every possible password and its message digest is called a *rainbow table*. If an attacker can compromise a computer system and capture any of the message digests that are stored on the computer, he simply finds the matching message digest that are stored on the computer in the rainbow tables to reveal the plaintext key or password used to produce the message digest.



However, another attack vector is *pattern detection*, in which the ciphertext is analyzed for patterns that might reveal the nature of the cryptographic keys. Some algorithms and cryptosystems show more patterns in their ciphertext, making these weaker but often faster than others. Additional manipulations can be combined with these systems to abstract the nature of the keys further, diffusing these patterns and making the cryptosystems stronger. However, these additional strengthening manipulations take time and processing power, resulting in a higher cost. Generally speaking, if a cryptosystem shows patterns in the ciphertext, that cryptosystem should only be used to protect small amounts of data, giving the bad guys only a small sample of ciphertext to analyze—small enough not to expose patterns and should not be used on very valuable data.

Basic substitution ciphers, described in the “Historical review of cryptography” section later in this chapter, lend themselves to pattern-based frequency analysis attacks. Transposition ciphers (also called permutation ciphers) contain the entire plaintext message in the ciphertext, and it is a simple process, often a game, to rearrange these letters or words in the correct order (into recognizable words or sentence patterns) and reveal the message. An attack on the patterns in ciphertext is typically much faster than the brute force attack but is not certain to be successful.



Not classically part of cryptanalysis, but a very successful technique for revealing keys and messages, is *social engineering*, in which the bad guy tricks a user into providing unauthorized access and revealing keys and messages. Social engineering is very often the fastest of the attacks on cryptography, but again, it does not represent the strength of the cipher or cryptosystems. Social engineering attacks identify the weaknesses in the human element in the information environment. These weaknesses can be reduced by providing security awareness training to all users, reinforcing the need to choose strong keys and passwords, and introducing users to the techniques of social engineers who intend to steal the users’ secrets, identity, and access.

NOTE FREQUENCY ANALYSIS ATTACK

Take any 10 pages of US English copy from any source—a newspaper, an encyclopedia, your favorite trashy novel—and count the number of times each character is used. The proportional usage of letters is very consistent, no matter what the source of the sample is, as shown in Figure 3-2. When a simple substitution cipher is used, attackers do exactly the same thing with the ciphertext—count the frequency of each character. They align that frequency of use with the frequency of use in a plaintext sample. This shows which plaintext characters to substitute for which ciphertext characters—typically with at least 80 percent accuracy. After that, simply eyeballing the results can tell them where corrections of the substitution need to be made to reveal the entire plaintext message.

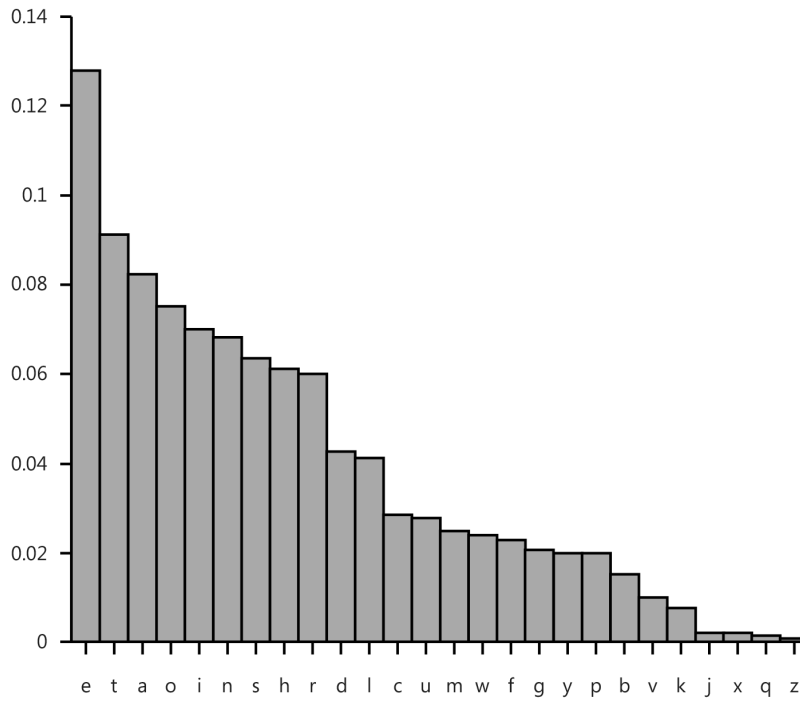
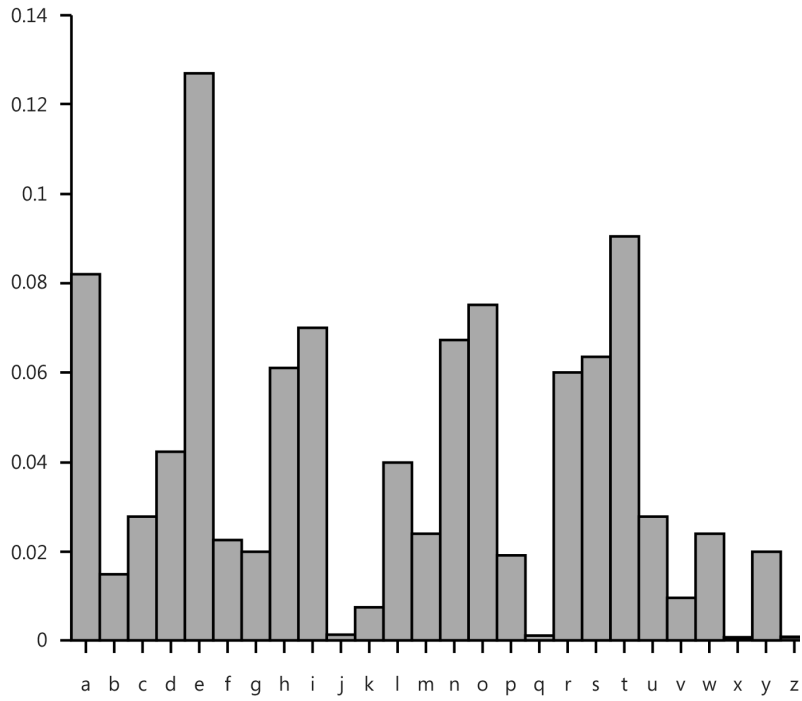


FIGURE 3-2 Frequency analysis of the use of US English letters

The strength of a cryptosystem—its work factor

In Chapter 1, in the discussion of the risk assessment and data classification processes, it was noted that the value of a particular data element (information asset) changes over time. For example, if a company has secretly developed a new product and is preparing to announce this new product next Monday, the value of this data would be very high today. However, Monday afternoon, after the announcement that grabs all the headlines has been made, the value of the information would be very low because it is now in the public domain.

If a competing company that has done no new product development were to learn of this information today, that competitor could make the announcement this Friday that it has developed the new product, to appear, untruthfully, to be the industry leader, superior to the legitimate company. Then, on Monday, when the legitimate company that had done the research and product development makes its announcement, that company appears to be nothing but a me-too follower in the industry and not the leader that it really is.

This example demonstrates more of a step function of how the value of data can change over time—in this case, very suddenly. Other data assets, such as your driver's license number, experience a more gradual change in value over time. Today, these numbers have value to you because they could be used to commit identity theft against you, but in a hundred years, this information will have very little relevance or value to anyone.

As the white-hat cryptanalysts work their magic and identify the work factor for a given cryptosystem, *they want the work factor of the cryptosystem to be substantially longer than the data the cryptosystem is protecting holds its value.* If your data will hold its value for 10 years, you should protect that data with a cryptosystem whose work factor is in the centuries (hundreds of years) or even greater—thousands of years or even millions or billions of years.

Why such an exaggerated time for the work factor? Because of a persistent nibbling away at the work factor called *Moore's Law*. Contemporary cryptosystems operate on computers. In 1965, Intel cofounder Gordon E. Moore predicted that the number of transistors packed into CPU chips (indirectly, the chip's processing power) should double every one to two years. His prediction has held true for more than 50 years of CPU chip development (1958–2008), as suggested by the graph shown in Figure 3-3. This means that if processing power used on our cryptosystems doubles every two years, the work factor of a particular cryptosystem will be cut in half every two years. This is why exceptionally long work factor values are needed for cryptosystems and why there is a constant evolution toward greater complexity with each new implementation of cryptosystems.



Microprocessor Transistor Counts 1971–2011 and Moore's Law

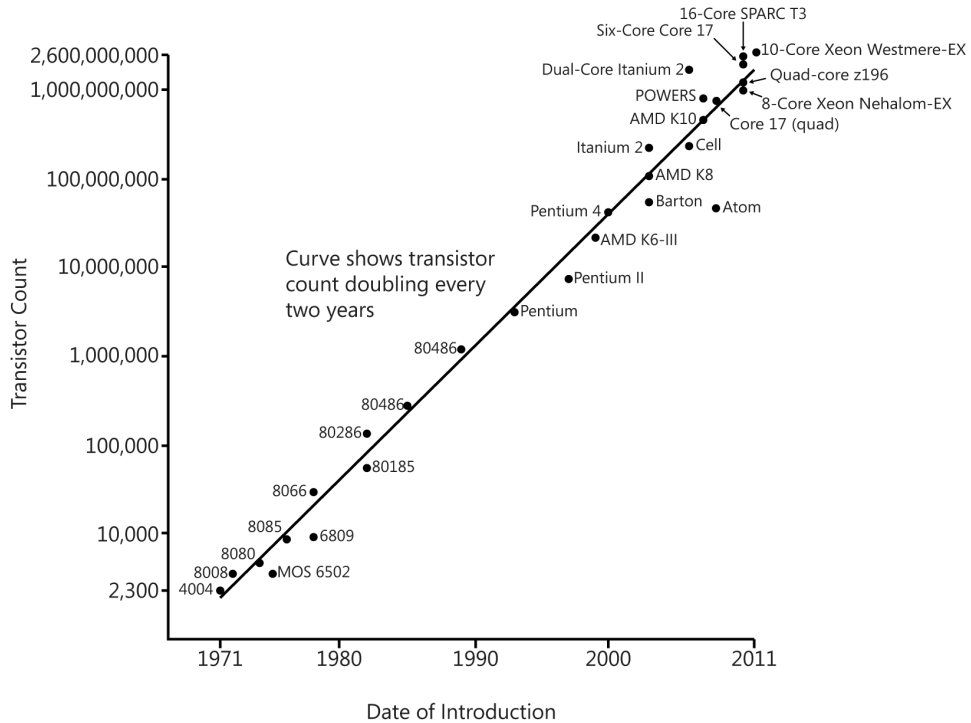


FIGURE 3-3 Moore's Law showing the number of transistors in a CPU chip doubling approximately every two years

NOTE MOORE'S LAW—LOOKING FORWARD

After the year 2015 or 2020, the increases in CPU power are predicted to slow to doubling only every three years.

Historical review of cryptography



Cryptography has been around for thousands of years. For almost all that time, there were only *symmetric key ciphers*, in which the key used to encrypt the message was exactly the same key used to decrypt the message. It was only when computers became commonplace (in the 1970s) that the computational horsepower was available to crunch numbers large enough to allow the invention of *asymmetric key ciphers*. *Asymmetric key ciphers* use two keys, a public key and a private key, that are mathematically related, and the content that one key encrypts can be decrypted only by the other key. The focus of this chapter will be the analysis of these symmetric and asymmetric key ciphers and cryptosystems.



EXAM TIP

Pay attention to the following features of each of these algorithms where they are relevant. These details will help guide you to the correct answers on the exam:

- Substitution
- Transposition
- Monoalphabetic
- Polyalphabetic
- The nature of the key
- Block cipher
- Stream cipher

Hieroglyphics: 3000 BC

Ancient Egyptians recorded events and concepts by using a defined set of characters or symbols to represent elements of the message being recorded. Hieroglyphics were readable only by those who were educated to read them. The intended recipient of the message had to be able to read hieroglyphics. This makes hieroglyphics a symmetric key *substitution cipher* because symbols were substituted to represent the meaning of the message, and the nature of the symmetric key was the education of the writer and the reader of the symbols.

The Atbash cipher: 500 BC

Atbash reverses the order of the characters in the ciphertext alphabet so that the plaintext letter A aligns with the ciphertext letter Z, the plaintext letter B aligns with the ciphertext letter Y, and so on. The characters in the plaintext message are substituted by using the characters on the ciphertext line, as shown in Figure 3-4.

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

FIGURE 3-4 The Atbash cipher

The plaintext message “CRYPTO IS COOL” is converted to the ciphertext message “XIBKGL RH XLLO.” The intended recipient of the message had to know the message was encoded using Atbash. The recipient simply performs the same character substitution. Atbash is a symmetric key, substitution, mono-alphabetic cipher, and the nature of the key is that both sender and recipient know the messages are encrypted using Atbash. It is a *monoalphabetic* cipher because it uses a simple one-line substitution function. (A *polyalphabetic* cipher, such as the Vigenere cipher, discussed later in this section, uses multiple lines of characters for its substitution function.)

Atbash is a very weak cipher because there is no variability in the substitution characters. The letter A is always substituted with the letter Z, the letter B with the letter Y, and so on. With practice, this cipher could easily become human readable.

The Scytale cipher: 400 BC

The Scytale cipher was implemented by wrapping a parchment, papyrus, or leather strap around a log of a specific diameter. Then the message was written across the log, one character per wrap of the leather strap around the log. When the leather strap was removed from the log, the letters appeared, one above the other on the strap, out of order. The strap would be given to a runner or horseman who carried it across the battlefield to the recipient. The intended recipient of the message had to know the diameter of the log used during the encoding process. The recipient would know to wrap the leather strap around the correct size of log. This would cause the letters to align and be once again human readable, as shown in Figure 3-5.

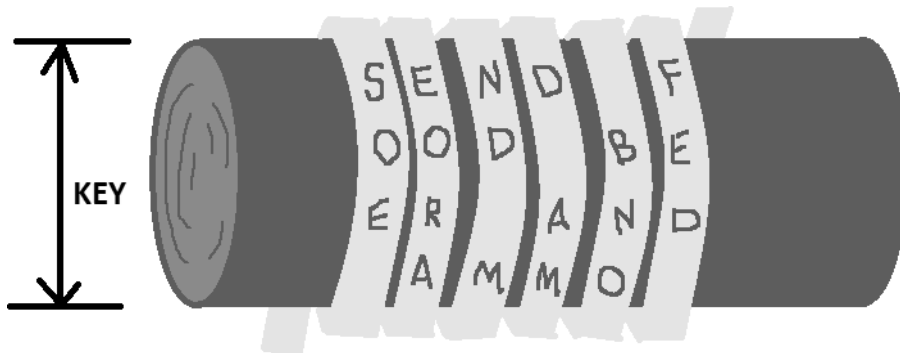


FIGURE 3-5 The Scytale cipher



The Scytale cipher is a symmetric key *transposition* cipher because the characters within the original plaintext are rearranged, or transposed, making them unreadable. The nature of the symmetric key is the diameter of the log used to encode the message.

The Caesar or Shift cipher: 100 BC

The Caesar or Shift cipher simply shifts the character set of the plaintext message by a specified number of letters and then uses the new character set as the substituted ciphertext characters. For example, if you choose a key value of 5, then you would substitute each character of the plaintext message with the character five places to the right in the alphabet, as shown in Figure 3-6.

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
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

FIGURE 3-6 The Shift cipher using 5 as the key value

The plaintext message, "CRYPTO IS COOL" in this case, is converted to the ciphertext message, "HWDUYT NX HTTQ." The intended recipient of the message has to know how many characters to shift for the message. The recipient simply reverses the direction of the shift to reveal the plaintext message. The Shift cipher added variability to the cryptosystem, allowing a different key from message to message.



Notice how the letters wrap around on the ciphertext line, beginning again at the letter A after the letter Z. This is referred to as *Modulo 26 addition*, which says, "After you reach the 26th character (which in the English alphabet is the letter Z), start at the beginning and get the next character from the beginning, the first character."

This is a symmetric key, mono-alphabetic, substitution cipher, and the nature of the key is the number of places to shift for the substituted ciphertext character. It is often said that the Caesar cipher always uses a shift (key) of 3, whereas the Shift cipher can shift by up to 25 characters. ROT-13 (rotate/shift by 13 places) is an example of the Shift cipher still in use today (mostly on UNIX and Usenet systems). In the English alphabet, with its 26 characters, if you rotate (shift) 13 places (encryption) and then rotate again 13 places (decryption), you end up where you started, with the cleartext message. The Caesar cipher can be referred to as a ROT-3 (rotate by 3 places) cipher.

A weakness in this type of cipher is that if any one plaintext letter/ciphertext letter mapping is known, the entire substitution mapping is now known. Another weakness of this type of cipher is that encrypting the message twice does not increase the strength of the ciphertext. Instead of decrypting with two shifts of 3, an attacker can successfully decrypt with a single shift of 6. This makes the shift cipher a member of the *group cipher* family of ciphers. Algorithms whose ciphertext is strengthened with multiple encryption processes (such as DES, described later in this chapter, in the section "Symmetric key algorithms and cryptosystems") are not group ciphers, and the ciphertext is strengthened by encrypting it multiple times.



An extension of this cipher is an *arbitrary substitution cipher*, in which the one plaintext letter/ciphertext letter mapping is randomized. The substitution letters on the ciphertext line have been randomly transposed. The symmetric key just became much more complex, but so did the ciphertext. With an arbitrary substitution cipher, if one plaintext letter/ciphertext letter mapping is known, only that one character mapping is known.

Cryptanalysis: AD 800

The first known recorded instance of cryptosystems cracking occurred in the ninth century by Al-Kindi, an Arabian mathematician in a manuscript entitled *A Manuscript on Deciphering Cryptographic Messages*. The document described the use of frequency analysis to attack contemporary symmetric key, mono-alphabetic substitution ciphers.

The Vigenere cipher: AD 1586

The Vigenere cipher used multiple lines of substitution characters to defeat frequency analysis attacks on ciphertext, making this the first recorded polyalphabetic cipher. In the Vigenere cipher, a string of characters would be chosen as the key, and those characters would be aligned with the characters of the plaintext message—for example, you could use the plaintext message “CRYPTO IS COOL” and a key of “BOBOVILLE.” The characters of the key would be repeated as necessary to cover the plaintext message characters, as shown in Figure 3-7.

Plaintext	C	R	Y	P	T	O		I	S		C	O	O	L
Key	B	O	B	O	V	I		L	L		E	B	O	B

FIGURE 3-7 Plaintext and key material for use with the Vigenere cipher

These characters would then be used in the Vigenere table, as shown in Figure 3-8, aligning the first plaintext character on the top horizontal line and the first key character on the leftmost vertical line.

		PLAINTEXT CHARACTERS																											
		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		
KEY CHARACTERS	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	CIPHERTEXT CHARACTERS	
	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		
	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		

FIGURE 3-8 The Vigenere cipher table

In Figure 3-8, every row and every column contains the entire alphabet, but the characters in each adjacent row or column are shifted by one character and apply the modulo 26 function. The 26 x 26 matrix is completed with the cell at the intersection of row Z (plaintext characters) and column Z (key characters) being filled with the letter Y. The first combination of plaintext C and key material B intersects at the substituted ciphertext character D; the next plaintext character R and the second character of the key material O intersect at the substituted ciphertext character F, and so on, to produce the ciphertext message shown in Figure 3-9.

Plaintext	C	R	Y	P	T	O		I	S	C	O	O	L	
Key	B	O	B	O	V	I		L	L	E	B	O	B	
Ciphertext	D	F	Z	D	O	W		T	D		G	P	C	M

FIGURE 3-9 Plaintext, key material, and ciphertext using the Vigenere cipher

As always, the intended recipient of the message has to know the symmetric key characters the sender used during the encryption of the message. The recipient locates the first character of the key in the leftmost vertical column and follows the horizontal line from that first character of the key to the right until she hits the ciphertext character. Following that line up to the top horizontal line reveals the plaintext character.

The Vigenere cipher added a notable increase in randomness to the ciphertext because it used multiple lines of substitution ciphertext characters, making this a relatively strong symmetric key, substitution, poly-alphabetic cipher.

The Vigenere cipher also introduces a critical concept that carries forward into today's cryptosystems. Suppose the key used was only a single character long. It would have to be repeated to cover all the characters of the plaintext message. This dramatically reduces the randomness of the ciphertext that the Vigenere cipher would otherwise provide. Further, suppose the key used was ABCD. This easily recognizable pattern or sequence would also dramatically reduce the potential randomness of the ciphertext. The Vigenere cipher introduced the concept that a stronger key would contain many characters, and those key characters would be highly randomized, a concept that remains intact on contemporary cryptosystems.

The Jefferson disk: AD 1795

The Jefferson disk included a set of 36 leather disks. The alphabet was stamped on the edge of each disk but in a randomized order on each disk. There was a hole in the center of each disk so it could be placed on a central axis or spindle. Each disk was uniquely numbered from 1 to 36 and was placed on the spindle in a specified order, which was the key for the message to be encoded or decoded.

The sender would then rotate the disks until the desired message was spelled out along a row of the letters on the edge of the wheels. The sender could then choose any other row of characters to be the ciphertext message to send to the recipient.

The recipient would rebuild his Jefferson disks in the correct order, rotate the wheels so one row of characters was the ciphertext message received from the sender, and then simply look for a row of characters that was a readable message elsewhere on the set of disks.

This system was the basis of the cryptography later used by the US Army from about 1922 until about 1945. The later evolution of this device the US Army used was called the M-94.

The Vernam cipher/the one-time pad: AD 1917

Named after Gilbert Vernam, who worked for AT&T Bell Labs, the Vernam cipher took the concepts of the Vigenere cipher to the full extent. Gilbert Vernam deduced that to strengthen the ciphertext against cracking, the number of characters in the key should be equal to the number of characters in the plaintext message to be protected, and those key characters should be highly randomized with no observable patterns. Each key should only be used once and then never reused, and, of course, the key must be kept secret. The Vernam cipher, which implements these characteristics, is also called the one-time pad. It is a symmetric key, poly-alphabetic, substitution cipher.

NOTE GILBERT SANDFORD VERNAM

Gilbert Vernam also invented the symmetric keystream cipher, a family of ciphers that will be reviewed in the "Symmetric keystream ciphers" section later in this chapter.

The Enigma machine: AD 1942

The Enigma machine was an electromechanical cryptographic machine based on a Hebern rotor-based system. Enigma used mechanical rotors to alter the pathway of electricity and light up ciphertext characters on the display. It was invented in Germany in the 1920s for commercial purposes. However, when World War II broke out, Nazi Germany relied on it heavily for its secure communications. Interestingly, just prior to the outbreak of World War II in 1939, a team of Polish cryptographers cracked the Enigma ciphers and secretly provided their findings to the French and British governments. This allowed the Allied forces to decipher many German communications and shorten the duration of the war in Europe.

The Enigma machine is a symmetric key, poly-alphabetic, substitution cipher system. Two models of the Enigma were built, one using a three-rotor system, and another using a four-rotor system (shown in Figure 3-10). Code names used by the Allied forces for the Enigma machine during the war were Triton and Shark.

The Japanese Red and Purple machines and the US Sigba machine are variations of the rotor-based Enigma machine.



FIGURE 3-10 The Enigma machine

Hashing algorithms: AD 1953

Hashing algorithms are an integral component of virtually all cryptography systems today. Although technically not cryptographic ciphers (no key, no way to reveal the original message from the hashed value), they are commonly used to provide integrity verification of data and authentication of one or more endpoints in a data transmission. The first hashing algorithm was invented by Hans Peter Luhn while he worked at IBM. The term *hashing* came from the chopping and mixing of the input data to produce the hashed output value. However, at that time, this functionality was viewed more as a mathematical curiosity than as something useful. This same Mr. Luhn invented the Luhn calculation used by virtually every credit card company to encode a validation check into the credit card number.

It was Robert Morris (the father of the man who released the infamous Morris worm [1988]) who, while working at the National Security Agency (NSA), recognized the importance of hashing algorithms in cryptography and converted the mathematical peculiarity into contemporary crypto-technology.

The Data Encryption Algorithm (DEA) and the Data Encryption Standard (DES): AD 1976



In the early 1970s, the US government recognized a need for establishing a standard for protecting sensitive but unclassified content as part of the *Federal Information Processing Standard (FIPS)*. After surveying the marketplace for ciphers, the government in 1976 selected the Lucifer algorithm submitted by IBM. Lucifer is a symmetric key block cipher, encrypting 128-bit blocks at a time and using a 128-bit key. As in virtually all contemporary symmetric key block ciphers, Lucifer uses multiple rounds of substitution and transposition to introduce confusion and diffusion into the ciphertext. This cipher was considered stronger than what was needed, so the National Institute of Standards and Technology (NIST) watered it down to a 64-bit block size and 64-bit key size to produce the Data Encryption Algorithm (DEA), used as the core of the Data Encryption Standard (DES) in FIPS. The DEA key is actually made up of 56 bits of key material and 8 bits of parity to produce the 64-bit key. DEA is a symmetric key block cipher. It performs 16 rounds of substitution and transposition to produce its ciphertext. DES is discussed in more detail in the “Symmetric key algorithms and cryptosystems” section later in this chapter.


Diffie-Hellman (or Diffie-Hellman-Merkle): AD 1976

After approximately 5,000 years of nothing but symmetric key cryptography and the weaknesses therein (described in the “Symmetric key algorithms and cryptosystems” section later in this chapter), computers became available to perform complex mathematics on unbelievably large numbers, and the very first asymmetric key algorithm was invented by Whitfield Diffie and Martin Hellman (1976). The Diffie-Hellman algorithm was invented to solve one of the major weaknesses in symmetric key cryptosystems: secure key distribution. In 2002, Martin Hellman recognized Ralph Merkle’s contribution to the asymmetric algorithm mathematics and proposed adding his name to this revolutionary and evolutionary advancement in cryptography. The Diffie-Hellman algorithm is discussed in more detail in the “Asymmetric key algorithms and cryptosystems” section later in this chapter.

NOTE CRYPTOGRAPHIC SERVICES

Although the Diffie-Hellman asymmetric key algorithm provides only one cryptographic service, secure key distribution, asymmetric key algorithms can potentially provide all five desirable cryptographic services in a strong manner. The five services are confidentiality, authentication, nonrepudiation, integrity, and secure key distribution.

RC4: AD 1987



Ron Rivest of Rivest, (Adi) Shamir, and (Len) Adleman (collectively known as RSA) produced a family of ciphers referred to as Rivest ciphers (hence the acronym RC, though some say that stands for “Ron’s Code”). RC4 (1987) is a symmetric *keystream cipher* that operates on binary bits, not characters. Stream ciphers, invented by Gilbert Vernam in 1917, encrypt a single binary bit at a time but introduce a high level of randomness in the resulting ciphertext. Although RC4 is a proprietary algorithm, it was leaked to the public domain in 1994. RC4 is by far the most prevalently used symmetric keystream cipher, finding its place in the original version of PPTP, in Wired Equivalent Privacy (WEP), and in Netscape’s SSL, web-based, secure channel. Symmetric keystream ciphers are discussed in more detail in the “Secure channels for LAN-based applications” section later in this chapter.

Triple DES (3DES): AD 1999

In 1997, a specialized computing system called Deep Crack with 1,856 crypto-cracking processor chips was able to crack DES in 96 days (that is a work factor of approximately three months). Shortly thereafter, another cryptanalysis team was able to crack it in about six weeks, then in a little over two days, and then in less than a day.

In 1999, DES was reaffirmed as the minimum standard, but the use of triple DES (3DES) was recommended. 3DES is three passes through DES by using two or three keys.

Today, relatively typical systems can routinely crack DES keys in about 4.5 days, with some weak keys revealed in less than a day.

Triple DES is discussed in more detail in the “Symmetric key algorithms and cryptosystems” section later in this chapter.

The Rijndael algorithm and the Advanced Encryption Standard (AES): AD 2002

With the observable weaknesses in DES and its relatively short keyspace, the US government needed a new standard. After surveying the marketplace once again, the government selected the Rijndael (pronounced rain-doll) algorithm by Belgian cryptographers Vincent Rijmen and Joan Daemen. The algorithm was incorporated into the Advanced Encryption Standard (AES) and in 2002 replaced DES and 3DES as the US FIPS standard for protecting sensitive but unclassified content. It is also approved by the NSA for protecting top secret data. AES is a symmetric key block cipher using a 128-bit block. AES can use key sizes of 128, 192, or 256 bits and performs 10, 12, or 14 rounds (respectively) of substitution and transposition.

Some say that AES is uncrackable. AES is discussed in more detail in the “Symmetric key algorithms and cryptosystems” section later in this chapter.

Other points of interest

Following are some additional ciphers and significant events in the history of cryptology.

- **Running key cipher** Both the sender and recipient have a shelf of books, exactly the same books in exactly the same order. This is the symmetric key. Ciphertext is created by locating the first word of the secret message in one of the books, identifying its location by writing the book number, page number, line number, and word number. Then the sender locates the second word in the secret message in another of the books and documents its location by using the four numbers described. The ciphertext is a page full of numbers. This is a symmetric key, substitution cipher, and the nature of the key is the common bookshelf of books. Its ciphertext might look like this (book, page, line, word):

4, 112, 17, 18, 11, 253, 6, 9, 9, 23, 31, 12

This cipher was also used to distribute symmetric keys securely for character substitution ciphers, with the ciphertext of this message identifying a line from a book to use as the symmetric key for the encryption and decryption of another message.

- **Concealment cipher** The concealment cipher is a symmetric key, transposition cipher in which the words or characters of the plaintext message are embedded in a page of words or characters at a regular interval. For example, the key could be that every fourth letter is a plaintext letter in the message. The other letters simply diffuse the meaning of the message, making it unreadable for those without the key. The key is the frequency of plaintext elements in the sheet of ciphertext. So the plaintext message "CRYPTO IS COOL" might look like this in ciphertext when the concealment cipher is used and the key is 4:

JBECPSLRUNXYAWUPNVSTIHGOEAZIKMWSTHJCSROYPLOVBWL PDU

- **AD 1815 to 1864: George Boole** A British mathematician largely regarded as the founder of the field of computer science and digital computer logic, George Boole studied the binary number system and developed a series of logical functions called *Boolean logic* and truth tables commonly used in integrated circuits, computers and, more relevantly, in cryptography. The binary Exclusive Or (XOR) function is the basis of a large proportion of contemporary symmetric key algorithms and cryptosystems.
- **AD 1883 - Kerckhoffs's principle** Auguste Kerckhoffs worked as a cryptographer for the French military and identified six design principles for cryptosystems. The most critical of these states the following (translated and paraphrased):

"The strength of a cryptosystem should not be in the secrecy of the algorithm but in the secrecy of the key."

This allows the algorithm to be published, studied, and scrutinized by the brightest minds on the planet in hopes of verifying the strength of the cryptosystem or to reveal its weaknesses. If the cryptosystem is strong, as long as the secret keys are maintained securely, unauthorized persons are unable to reveal the protected messages or reveal the nature of the keys.





- **AD 1915 to 1990 - Horst Feistel** This German-born physicist, mathematician, and cryptographer moved to the United States in 1934 and developed the Feistel Network while working at IBM. The *Feistel Network* has been the foundation of many symmetric key block ciphers, including DES, 3DES, RC5, RC6, IDEA, Blowfish, Twofish, Lucifer, CAST, GOST, MARS, and Skipjack. A key feature of the Feistel Network is that the algorithm is most often identical when performing the encryption and decryption functions. This allows developers to use the same code for the two processes with only minor adjustments (key scheduling). The Feistel Network includes an iterated cipher and introduced a popular and often used *round function*.



Quick check

1. What type of algorithm is the Vernam cipher?
2. Mixing and moving the characters of a plaintext message around to produce ciphertext describes what type of cipher?

Quick check answers

1. A symmetric key, poly-alphabetic, substitution cipher
2. A transposition cipher

Cryptographic keys

Cryptography relies on keys. In symmetric key cryptography, the key that is used to encrypt the content is the same key that is used to decrypt the content. The sender and the recipient must each have a copy of the key to perform the encryption and decryption process, so these two copies must be identical.

In asymmetric key cryptography, often referred to as *public key cryptography*, a pair of related keys is used. One key is a private key, created and stored securely and never shared. The mathematically related key in the key pair is the public key. These keys are related such that what one key encrypts, only the other key can decrypt. If you encrypt using the private key, only the public key can decrypt the content. If you encrypt using the public key, only the private key can successfully decrypt the content.



Key management becomes a core component of any cryptosystem. Key management includes the creation, distribution, storage, archiving (escrow), recovery, and disposal of the key material. Moreover, this must all happen securely so that attackers can never access your keys.

Key creation



In contemporary cryptosystems, cryptographic keys are generated on a computer system by a specialized kernel mode process called a *cryptographic service provider (CSP)*. Different CSPs produce different types of keys, as requested by the crypto-enabled application. The keys are strings of binary bits.

Key length

Symmetric keys are typically 64 bits to 512 bits long. Asymmetric keys are typically 768 bits to 4,096 bits (4 kilobits) long, with the current recommendation at 3,072 bits (3 kilobits) long. The key length used is typically defined by the developer of the crypto-enabled application and the algorithm or cryptosystem chosen.



Generally speaking, the longer the length of the key, the stronger the cryptosystem. Specifying a key length is very much like the childhood game of “I am thinking of a number between 0 and 9. Guess which number I am thinking of.” As the range of the possible chosen numbers increases, it becomes more difficult for the bad guys to guess which secret number you have chosen. Your key is a number chosen from this range of possible numbers, otherwise called the *keyspace*. Key lengths are defined by the number of binary bits. The range of possible numbers to choose your key from is directly related to the number of binary bits in the key, following the function 2 to the power of N , where N is the number of bits in the key. So a 64-bit key has a keyspace of 2 to the power of 64, or 18,446,744,073,709,551,616 possible choices of the symmetric key used. Your chosen key is a number somewhere between 0 and 18,446,744,073,709,551,615. The bad guy has to guess which number you chose. Then she can decrypt your messages. This is considered to be such a weak key length that it is not recommended. Table 3-1 shows the keyspace for some standard key lengths.

TABLE 3-1 Keyspace calculation

Bits	Keyspace - approximate	Cryptosystem
2^{64}	18 with 18 zeros behind it	SYMMETRIC
2^{128}	34 with 37 zeros behind it	SYMMETRIC
2^{256}	11 with 76 zeros behind it	SYMMETRIC
2^{512}	13 with 153 zeros behind it	SYMMETRIC / ASYMMETRIC
$2^{1,024}$	18 with 307 zeros behind it	ASYMMETRIC
$2^{2,048}$	32 with 615 zeros behind it	ASYMMETRIC
$2^{3,072}$	58 with 923 zeros behind it	ASYMMETRIC
$2^{4,096}$	10 with 1,232 zeros behind it	ASYMMETRIC
$2^{15,360}$	66 with 4,622 zeros behind it	ASYMMETRIC
$2^{32,768}$	14 with 9,863 zeros behind it	ASYMMETRIC

In May 2011, in SP800-57 (Part 1, Rev. 3), NIST recommended a minimum symmetric key length of 64 bits for 3DES using three keys (also called 3DES or TDES) and up to 256 bits for AES, using a single key. They recommend that asymmetric public keys should be a minimum of 2 kilobits (2,048 bits) long, up to as many as 15,360 bits. These minimums are currently expected to remain viable until the year 2030.

Key distribution

For the several thousand years of symmetric key cryptography, key distribution was a problem. If you had to get a sensitive message to a recipient in a different location, and the space between you was potentially hazardous, where an attacker might try to steal your data, you should encrypt your sensitive data. Right? But because you only had symmetric key cryptography, you had to get the recipient a copy of the symmetric key first. How would you get a copy of a symmetric key to that recipient securely? If you could get the key to your target securely, why not just tell the recipient the message by using that secure communications channel? Most often, the answer was that there was no secure way to get the key to the recipient unless you knew in advance that you would need this secure communication channel; thus, you would share a symmetric key prior to needing it, when you were in close proximity to one another. In other words, you needed a prior association with the recipient and had to anticipate the future need for secure communications.



If you had not anticipated this need, you had to have some alternative communications channel that might not be monitored by the bad guys. This is referred to as an *out-of-band* communications channel. Today, you might pick up the phone and tell the recipient that the password is BOBOVILLE. Or you might copy the key to a thumb drive and carry it (sneaker-net it) to the recipient instead of sending the key across the potentially hazardous network.

Symmetric key cryptography creates the problem of needing some secure mechanism to get a copy of the symmetric key to a recipient.

Today, Diffie-Hellman and the family of asymmetric algorithms, with their public keys and private keys, not only do not have this problem but finally solve this five-thousand-year-old problem by providing secure key distribution services. The way this works is described in detail in the “Asymmetric key algorithms and cryptosystems” section later in this chapter.

Secure key storage

Contemporary information systems exist on computers; therefore, contemporary cryptosystems exist on computers too. Typically, it is the responsibility of the operating system to provide a secure storage mechanism and location for cryptographic keys. Most operating systems establish and protect various key stores to achieve this. Because the keys are usually generated on the system by the operating system, it becomes natural for the operating system to store the keys securely, as needed.

Quantities of keys

As the number of participants (users) in a cryptosystem increases, you should consider the number of keys to be created, distributed, and stored for the life of the cryptosystem. Even very basic symmetric key cryptosystems require many keys, and every copy of every symmetric key must be protected for the life of the cryptosystem. The number of keys required can be determined by using the following formula, where N = the number of users in the cryptosystem:

$$(N \times (N - 1))/2$$

Table 3-2 shows the number of unique symmetric keys required as the number of users in a cryptosystem increases.

TABLE 3-2 Symmetric key requirements

Users	Number of keys
5	10
10	45
100	4,950
1,000	499,500

As you can see, the number of keys rises dramatically as the number of users in the symmetric key cryptosystem increases, and 1,000 users is not at all uncommon in today's IT infrastructure. Nevertheless, this isn't the worst part. Two copies (so double these numbers) of each of these keys must be protected for the life of the cryptosystem. Otherwise, there is a breach of security, and secrets will be lost.

Asymmetric key cryptography provides a distinct improvement in this area. In a basic asymmetric key cryptography, only two keys are required per user, providing the following formula, where N = the number of users in the cryptosystem:

$$2N$$

Table 3-3 shows the number of unique asymmetric keys required as the number of users in the cryptosystem increases.

TABLE 3-3 Asymmetric key requirements

Users	Number of keys
5	10
10	20
100	200
1,000	2,000

This shows a dramatic improvement in reducing the number of keys required in the cryptosystem, and the good news continues. Only the private key in the public/private key pair must be kept private. The public key can be shared with anyone without any risk at all. So only half of the asymmetric keys must be protected for the life of the cryptosystem.

Key escrow (archival) and recovery

One of the common responsibilities of IT personnel is to ensure the recoverability of all the organization's valuable information assets. So if a user has encrypted content and somehow loses his copy of the decryption key, it is IT's responsibility to be able to recover (decrypt) that content, making it usable and recovering its value. One technique used to accomplish this is to escrow all decryption keys. You can imagine that this repository of all the decryption keys for an organization would be a juicy target for the bad guys, so you recognize the need for the utmost security to protect these escrowed decryption keys. If someone could get into this repository and gain access to the archived keys, he might be able to decrypt all protected content within the organization and steal the identity of every user participating in the cryptosystem.



Many companies use specialized *hardware security modules (HSMs)* to store and retrieve these escrowed keys securely. These HSM systems typically detect and prevent tampering by destroying the key material if tampering is detected (kind of like in *Mission: Impossible*).



Recovery of these archived keys is usually considered to be too much authority for a single individual, so these HSM systems are typically designed to support dual control, or an *M-of-N function* requiring several people to come together and agree on the recovery of even a single key. Typically, the system provides several recovery keys, which might be like USB thumb drives. The HSM manufacturer might provide, for example, eight recovery keys. Then, based on company policy and the HSM system's configuration, it might require five of the eight recovery agents with their recovery keys to come together and agree on the recovery of any escrowed keys.

NOTE CRYPTO CONTENT JOURNALING

Content journaling is another approach to solving the problem of recovering encrypted content. With content journaling, the cryptosystem is configured to send a copy of all encrypted content to a highly secured, service-type user account, often called a recovery agent account. The cryptosystem is set up to allow this recovery agent user to decrypt all content. Recognize the technology requirements and the storage capacity issues that might arise from this type of solution in addition to the massive sensitivity (value) of this repository and the need for serious protection and auditing. No user should ever log on to this account unless authorized for the strict purposes of data recovery of specified content.

Key lifetime or the cryptoperiod

Keys should be changed regularly, just as most systems require you to change your password every 60 or 90 days. Company policy should define how frequently these passwords and the other keys used in the organization must be revoked and replaced with new keys. This is called *key rotation*, and the useful lifetime of a key pair is called the key's *cryptoperiod*. Keys used on a user's digital certificates often are good for 1 or 2 years. Keys on the certification authority servers are often good for 5, 10, or even 20 years. However, some keys, such as those used on a high-security VPN channel, might be changed as frequently as every few milliseconds (thousandths of a second), and other cryptoperiods range between these extremes.

It takes administrative effort, time, and system processing to rotate keys, so this must be balanced—as always, cost justified—against the need for increased security. Policy should dictate the cryptoperiod for the various types of keys used within a cryptosystem. Following are a few of the issues that would indicate the need for a short cryptoperiod:

- Weak algorithm(s) being used and the implementation details
- A high value assigned to the data that is being protected with the keys
- A lower level of trust for the entity (or entities) using the keys
- Use of keys or ciphertext in very hostile environments
- A high frequency of access to the keys
- A large quantity of data to be protected with keys (a larger sample of ciphertext)

The following lists some terms you should become familiar with; these provide an indication of the cryptoperiod of a key:

session key = short-term key = temporal key

secret key = persistent key = static key = long-term key

Session keys, also called short-term or temporal keys, have a relatively short cryptoperiod, typically the duration of a session. This can be as short as a few milliseconds to as long as a day or so.


Secret keys, also called persistent, long-term, or static keys, have a comparatively longer cryptoperiod. This can be as short as a day or a few days to as long as a few years or even decades.

In 2011, NIST recommended a cryptoperiod of less than or equal to two years for most cryptographic keys used to protect sensitive but unclassified content.


When the cryptoperiod expires, or when the administrator of the cryptosystem has just cause, the keys a user uses in a cryptosystem can be revoked and destroyed. These expired or revoked keys should not be used again for additional cryptographic functions other than to recover existing encrypted content if necessary.

Initialization vectors


A common issue to be concerned about with any cryptosystem is to what extent the nature of the encryption key presents itself in the resulting ciphertext. This is one of the most common attack vectors. If all things are working correctly, the only thing the bad guys have access to is the ciphertext as it flows between trusted entities over relatively untrusted network segments. If the ciphertext shows patterns, it might lead to insights into the nature of the encryption key, which leads to cracking the key.

 One technique used to abstract the nature of the encryption key further in the resulting ciphertext is the addition of one or more *initialization vectors (IVs)* in the algorithm or in the implementation of the cryptosystem. This IV is a *nonsecret variable* that affects the processing of the plaintext data during encryption. Generally, the longer the IV value is, the greater the randomization and the stronger the resulting ciphertext. The goal is to make the ciphertext more randomized, diffusing potential patterns and hiding any insights or clues into the nature of the encryption key used to produce the ciphertext.


The IV must be shared with the recipient and can be sent in plaintext. If the bad guys see the IV and the ciphertext only, they gain less insight into the nature of the encryption key than without the use of an IV.

 Other terms that you might see that refer to values that operate like an IV are *salt*, *seed*, or *nonce*.

Hashing algorithm/message digest

 Hashing algorithms, also called message digests, are primarily used for the verification of the integrity of information at the time of use of the information. The hashing algorithm is applied to the message to produce a hash value. The hash value acts as a *fingerprnt* of the message and is initially calculated at the time of creation of a message or at other times when the information in the message is known to be complete and accurate. Then at a later time or place, when the information in the message is about to be used, the calculation is performed again and compared to the original hash value. If the two values are identical, it is concluded that the message says exactly what it did at the time it was known to be good.

Hashing is also used in cryptosystems to provide authentication services, providing some level of trust that the sender is who he claims to be.

 Hashing algorithms can accept any size of message as input and produce a fixed-size hash value (also called a *message digest*) output. Messages on computers are stored as a contiguous string of binary bits. In a simplified description, the hashing algorithm breaks that long contiguous string of binary bits into smaller chunks of binary bits. The algorithm then treats those chunks of binary bits as numbers and runs mathematical and logical functions on those numbers to produce a fixed-length hash value output as the answer.

A hashing algorithm will produce the same answer (hash value) on the same message every time as long as none of the binary bits that make up the message have been changed.

If any binary bits in the original message, even one, are changed between the original hash value calculation and the time-of-use calculation, the chunk of binary bits that contains the changed bit or bits results in a different number, and the mathematics performed on those different numbers will produce a different answer, a different fingerprint. This difference in the before (creation) and after (altered) fingerprints identifies that a change has occurred in the data. This is used commonly in cryptosystems to identify a violation of the integrity of the original data, called integrity verification.

The greater the number of bits in the hash value output from a hashing algorithm, the more specific the fingerprint of the message. More bits in the hash value, therefore, make the hashed message more difficult to alter yet provide the same hash value to make the altered message believable and trusted (called spoofing a message). Table 3-4 shows several commonly used hashing algorithms and their hash value output bit lengths.

TABLE 3-4 Hashing algorithms and their output lengths

Algorithm	Output bits	Notes
PARITY	1	The crudest of all hashing functions. Invented in ancient Greece.
CYCLIC REDUNDANCY CHECK (CRC)	32	Appended as a trailer to Ethernet packets to detect corruption, not tampering. 1961, 1975.
MESSAGE DIGEST v2 (MD2)	128	By Ron Rivest. Optimized for 8-bit computers. Rare but still used in some public-key infrastructures (PKIs). 1989.
MD4	128	Ron Rivest. Used in NTLM authentication. 1990.
MD5	128	Ron Rivest. Flaws found. 1991.
MD6	512	Ron Rivest. Flaws found. 2008.
SECURE HASHING ALGORITHM v1 (SHA1)	160	Developed by NIST. FIPS PUB 180-1 Standard 1995–2002.
SECURE HASHING ALGORITHM v2 (SHA2)	256-512	Developed by NIST. FIPS PUB 180-2 Standard 2002.
SHA3	Arbitrary	NIST has chosen a base algorithm called Keccak for the upcoming SHA3 standard. SHA3 will use a mathematical sponge construction by which input data is absorbed into the algorithm. Examples currently show output sizes ranging from 244 bits to 512 bits, but the algorithm's functionality allows an expansion of this range.
HAVAL	128-256	128-bit, 160-bit, 192-bit, 224-bit, and 256-bit output options. 1992.
RIPEMD	128-320	128-bit, 256-bit, and 320-bit output options. 1996.
TIGER	192	128-bit, 160-bit, and 192-bit output options. 1995.
WHIRLPOOL	512	ISO Standard. Vincent Rijmen (co-inventor of Rijndael used in AES). 2000.

The output from the hashing algorithm, called the hash value or message digest, is typically written in hexadecimal characters. For example, the SHA1 (160-bit) hash output for the input P@ssword is:

```
9e7c97801cb4cce87b6c02f98291a6420e6400ad
```

SHA-256 hash output for the input P@ssword is:

```
28efb68dcb507ecd182bead31e4e2d159b0f9185861d1ebfe60a12dfb310300
```

The same input should *always* produce the same hash output when the same hashing algorithm is used. Several websites will calculate the hash value for your input. Feel free to try some of them. You should get the same output values when using P@ssword as the input.

Hashing algorithms perform a one-way function. They can take any size of input, 10 bytes or 10 terabytes, and produce a fixed-length output. In the case of SHA1, this would be a 160-bit output hash value. Although this is an accurate fingerprint of the original 10-terabyte message, you certainly cannot convert the 160-bit hash value back into the original 10 terabytes of data.

Hashing algorithms do not use any kind of key as encryption and decryption algorithms do. Although hashing algorithms are commonly referred to as cryptographic hashing algorithms, because of the lack of a key and their one-way functionality (being unable to reveal the meaning of the original message from the hash value), hashing algorithms are, strictly speaking, *not* cryptographic functions.

Hashing algorithms, like all other algorithms, must be designed well to be usable. The more bits there are in the hash value output, the more specific the fingerprint of the message is and the stronger the hashing algorithm is. The algorithms must provide good strength versus their performance (time required to process). They must have a high avalanche effect; in other words, it should only take a little change at the top of the mountain (the message) to cause a huge change at the bottom of the mountain (the hash value output). The hash value for the message "IOU 10 beers." should be very different from the hash value of the message "IWO 11 beers." Equally important is for the hashing algorithm to be resistant to *collisions*. A collision is when two different messages produce the same hash value output—like two people with the same fingerprint—and this is a bad thing. The lower the frequency of collisions for an algorithm, the stronger the algorithm.



Attacks on hashing algorithms

The bad guys want to alter your data to gain some sort of benefit, usually to steal money. They want to alter the inventory count so they can take your inventory without detection. They want to inject malicious code into a downloadable application or device driver to compromise your computer. They want the contract to read that you must pay them \$10,000 instead of \$1,000.

a strong initialization vector, the use of different types of algorithms together within a cryptosystem, and the addition of hashing algorithms to provide integrity-verification and authentication capabilities. Such a system is called a hybrid cryptosystem.

The addition of these randomizing functions, which will degrade performance, should be balanced with the need for stronger security. This balance must be defined by the policies of the organization that specify the level of protection required for specific classifications (values) of data. Provide an appropriate level of security and not more. Otherwise, you are paying too much for security.

This last bullet point is debatable. Many believe that the world, including potential attackers should also know the details of the algorithm. Kerckhoffs's principle says to publish your algorithms and let them be tested under fire by the brightest mathematicians and cryptographers on the planet. If no one can find flaws or weaknesses, and you can verify a sufficiently long work factor, then you have a pretty good level of trust that the algorithm or system is strong. The strength of your cryptosystem should come from the secrecy of the keys, not the secrecy of the algorithm.

Very often, this is not done with algorithms and cryptosystems for government use, but it is common in the public sector.

NOTE STRONG CIPHERTEXT

Strong ciphertext should not provide any insight into the nature of the encryption key that was used to create the ciphertext. Furthermore, no patterns within the ciphertext should provide any insight into the nature of the encryption key. Strong ciphertext should not compress well because compression algorithms identify patterns and then substitute smaller markers for the multiples of the pattern to reduce the file size. If the ciphertext is highly randomized, there should be no multiples of patterns.

**EXAM TIP**

Different types of ciphers and cryptosystems provide different cryptographic services (can you list the five desirable cryptographic services?) at varying levels of strength. Understand which services and levels of strength of those services each cipher and cryptosystem can provide.

Symmetric key algorithms and cryptosystems

As stated previously, a cryptographic cipher is a manipulation process that transforms organized, readable information into unreadable content and can be used later to recover the meaning of the information by making it readable again. Algorithms introduce confusion or diffusion, often by performing substitution, transposition, or mathematical functions on the data.

Cryptographic algorithms fall into two major categories:

- Symmetric key algorithms
- Asymmetric key algorithms

This section and the following sections cover these algorithms in depth, starting here with symmetric key algorithms.

The term *symmetric* implies that the key used to encrypt the content is the same as the key used to decrypt the content. This means that the sender and the recipient must each have a copy of the symmetric key.

Symmetric key cryptography is *fast* compared to asymmetric key cryptography. Depending on the specific ciphers used and the other features of the cryptosystem, symmetric key cryptography is estimated to be somewhere between 100 times faster and, in many cases, more than 1,000 times faster than asymmetric key cryptography. So virtually all bulk encryption, when the volume of content is undefined or known to be more than a few kilobytes (KB), uses symmetric key cryptography.

Today, information systems rely on cryptography to provide five critical cryptographic services:

- Confidentiality
- Authentication
- Nonrepudiation
- Integrity
- Secure key distribution

Symmetric key ciphers provide only one of the five desirable cryptographic services strongly, confidentiality. Encryption and decryption processes that use symmetric keys perform very well and can produce *strong ciphertext for strong confidentiality*.

Symmetric keys can be used to provide weak authentication. If you receive a message that is successfully decrypted by using a symmetric key you share only with BoBo, then it is very likely that BoBo sent you the message because no one else but you and BoBo should have a copy of the key that encrypts the message. Therefore, you have a level of trust that the message came from BoBo. This is authentication. However, because there are two keys (you have one and BoBo has one) that could have been used to encrypt the message so that your copy of the symmetric key successfully decrypts the message, it can never be proven which key was used to produce the ciphertext. This makes symmetric key cryptography a *weak source of authentication*.

Because the authentication services provided by symmetric key cryptography are weak, BoBo could deny sending the message, and there is no way ever to prove he is lying. So symmetric key cryptography *does not provide nonrepudiation*. To provide nonrepudiation, the cryptosystem would need to prove with certainty that BoBo was the source of the encrypted message. With this provability, BoBo could not deny being the source.

If you add a hashing function to a symmetric key cryptosystem, it can be used to prove that a message has not been altered since the time it was sent. This is integrity validation. However, because you cannot be certain who sent you the message (weak authentication), you cannot trust the message any more strongly than you trust the authentication. Symmetric key cryptography provides *weak integrity validation*.

Finally, symmetric key cryptography not only does not solve the problem of needing some secure mechanism to distribute symmetric keys over nonsecure channels, it creates the problem. *It does not provide secure key distribution services; it causes the problem.*

The following list of services shows what symmetric key cryptography can provide and at what level of strength:

- **Confidentiality** Strong
- **Authentication** Weak
- **Nonrepudiation** No
- **Integrity** Weak
- **Secure key distribution** No, the use of symmetric keys causes the problem

Users of a cryptosystem typically only consciously recognize two services they need. These are commonly referred to as *signing* and *sealing*.



In a symmetric key cryptosystem, *signing* a message provides weak authentication and weak integrity validation services (with the addition of a hashing algorithm). Nothing else. Recognize the need to distribute the symmetric key securely to the recipient.



In a symmetric key cryptosystem, *sealing* a message provides strong confidentiality services. Nothing else. Recognize the need to distribute the symmetric key securely for decryption to the recipient.

A message can be signed, sealed, or both at the discretion of the user, based on policy, or as imposed by the cryptosystem.

In addition, recall the issues regarding the number of keys required in a symmetric key cryptosystem, described earlier in this chapter: $(N \times (N - 1))/2$.

Symmetric key algorithms operate by performing substitution, transposition, or both functions on the data. Substitution replaces a plaintext character with a ciphertext character and introduces confusion in the ciphertext. Substitution ciphers are sensitive to the frequency analysis attack. Transposition mixes and relocates plaintext characters in the ciphertext, introducing diffusion in the ciphertext, as in the popular game that gives you jumbles of the letters of words for you to figure out what the secret message is. Transposition ciphers leave all the data from the original, sensitive plaintext message in the resulting ciphertext message. When used alone, both substitution and transposition ciphers present vulnerabilities, but when they are used together, as is common in contemporary symmetric key cryptosystems, those vulnerabilities are greatly reduced.

Present-day information systems store and process data on computers, and therefore, the symmetric key cryptographic systems operate on the binary bits that represent the data we use and understand.

Symmetric key algorithms come in two types: block ciphers and stream ciphers.

Symmetric keystream ciphers

Invented by Gilbert Vernam in approximately 1917 at AT&T Bell Labs, stream ciphers encrypt and decrypt a single bit at a time and operate like a one-time pad. The algorithms typically use a symmetric key and an IV as inputs, a *pseudo-random number generator (PRNG)*, and the symmetric keystream algorithm itself to produce a *keystream* of binary bits. This keystream output should be made up of long periods of nonrepeating sequences and provide bits in a statistically unbiased manner (over a sampling, an equal number of 1s and 0s). Further, although the keystream is a byproduct of the key, the keystream should not show any linear relationship to the key. The initialization vector and the PRNG help avoid this issue.

The binary XOR function is then used on the keystream with the bits of the plaintext to produce the ciphertext. This is shown in Figure 3-11.

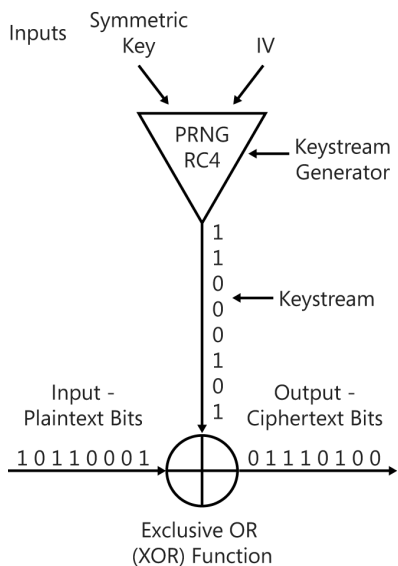


FIGURE 3-11 The keystream generator

George Boole, the nineteenth-century British mathematician, was mentioned earlier in this chapter. This is where his magic comes into play—inside the Exclusive Or (XOR) function. Boole developed a series of binary logic functions called Boolean logic, and their resulting *truth tables*: functions such as AND, OR, NAND, NOR, and XOR, to name a few. These functions are the basis of digital circuitry and are heavily used today. The XOR logic function lends itself beautifully to cryptography. With this *XOR function*, if two binary bits (A and B)

are provided as input, the result (R) will be true (a binary 1) if one bit or exclusively the other bit is a 1. Otherwise, the result is false (a binary 0), as shown in Figures 3-12 and 3-13.

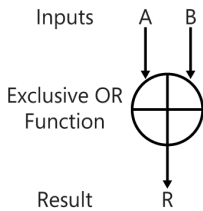


FIGURE 3-12 The Exclusive OR (XOR) function

EXCLUSIVE OR (XOR)			
Input			Result
A	B	=	R
0	0	=	0
0	1	=	1
1	0	=	1
1	1	=	0

FIGURE 3-13 The XOR truth table

When this function is used on binary plaintext data as input A and binary key material as input B, it performs a reversible, symmetric key substitution process to produce ciphertext (R). The XOR function is a core component of the stream cipher and is the basis of the S-box function used in block ciphers, described in the “Symmetric key block ciphers” section later in this chapter. Figure 3-14 demonstrates the XOR encryption process and decryption process.

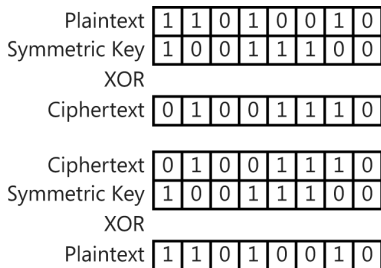


FIGURE 3-14 Encryption and decryption using the XOR function

Notice that the ciphertext binary string is different from the plaintext binary string. The plaintext bits have been substituted by the ciphertext bits. This ciphertext can now be securely stored or transmitted over untrusted channels because it does not reveal the original information. The recipient, who must also know the same symmetric key, can decrypt the ciphertext into the original plaintext message. Notice that the two keys are identical (symmetric). The bad guy, who might see the ciphertext message, does not know the symmetric key, and therefore, cannot convert the ciphertext back into the original, readable message. Confidentiality of the message has been provided.

Work through your own sample of plaintext and key material by using the XOR function (encryption) to produce ciphertext and then recover the plaintext from the ciphertext (decryption). You might be a cryptographer now!

NOTE WHY PSEUDO-RANDOM?

True randomness is a difficult beast to create. True random number generators use unpredictable processes in nature to generate randomness, such as wind noise or cosmic background radiation. Inside a computer, these elements are unavailable, so the computer system itself must be the source of the randomized values. Unfortunately, although the system can get very close to being truly random, because of the relatively small set of variables as inputs available within a computer system, small and subtle patterns might exist in their output.

Symmetric keystream ciphers produce stronger ciphertext than block ciphers because of the reduction in potential patterns within the resulting ciphertext, but they are generally considered about 1,000 times slower than block ciphers.

Because they encrypt a single bit at a time, and because of their relative slowness when compared with block ciphers, stream ciphers are better used to encrypt small amounts of data, often single bits or single bytes (8 bits) at a time. When block ciphers are used to encrypt small amounts of data, specifically smaller than their block size, they must generate padding bits to fill the block. This padding often presents patterns and further weakens the strength of the block cipher.



Most symmetric keystream ciphers rely on a function well-performed on an integrated circuit chip (hardware) called a *Linear Feedback Shift Register (LFSR)*. Most stream ciphers *tend not to code well* in applications without that hardware, and the code without the use of the LFSR chip does not process efficiently on CPUs. Therefore, it is said that *symmetric keystream ciphers are best implemented in hardware*. This should be compared to the implementation of symmetric key block ciphers, described in the next section.

Following is a list of symmetric keystream ciphers to become familiar with:

- RC4
- A5/1, A5/2
- Rabbit
- FISH
- SNOW
- SOBER, SOBER-128
- ISAAC
- MUGI
- Scream



EXAM TIP

Only one symmetric keystream cipher will be examined in detail in this chapter, RC4. Be aware of the others, not so much to know details about what they are, but as to know what they are not. They might be used as distractors on the exam. Just know that they are symmetric keystream ciphers.



In approximately 2004, the European Union established a project called *eSTREAM* to evaluate new stream ciphers, hoping to accelerate their development and adoption as international standards for cryptography.

RC4

Designed in 1987 by Ron Rivest, RC4 is the most prevalently used symmetric keystream cipher. RC4 can use key sizes ranging from 40 bits to 256 bits. Although it is proprietarily owned by RSA, RC4 was leaked into the public domain through Cypherpunks in 1994 and then out into the wild from there. Because the term *RC4* was trademarked by RSA, it might also be referred to as ARCFOUR and ARC4 to avoid trademark issues with RSA. It is used in the original implementation of PPTP, Secure Shell (SSH), Remote Desktop Protocol, (RDP), WEP, Wi-Fi Protected Access (WPA), SSL, and Transport Layer Security (TLS). It is believed that RC4 became so popular due to its speed and its simplicity of implementation. RC4 does not require the use of hardware LFSRs, unlike most stream ciphers.

Symmetric key block ciphers

Block ciphers encrypt and decrypt a block of data at a time, making these ciphers the fastest of all. However, these blocks of ciphertext tend to be the most revealing of patterns and therefore require additional consideration in their use. It would be unacceptable to choose a block cipher because of its excellent performance (speed) but allow the decryption key to be revealed in its ciphertext. The next section looks at multiple modes of operation for block ciphers that progressively improve the randomization in the ciphertext, further abstracting the nature of the encryption key.

Virtually all contemporary symmetric key block ciphers use both substitution and transposition functions. The substitution box, or S-box, performs an XOR function on a binary bit of plaintext and a binary bit of key material, a block of it at a time. An S-box function and a transposition function together is called *one round* of cryptographic processing, as shown in Figure 3-15. Different algorithms perform different numbers of rounds to produce their ciphertext. For example, DES performs 16 rounds of substitution and transposition. The International Data Encryption Algorithm (IDEA) performs eight rounds of substitution and transposition.

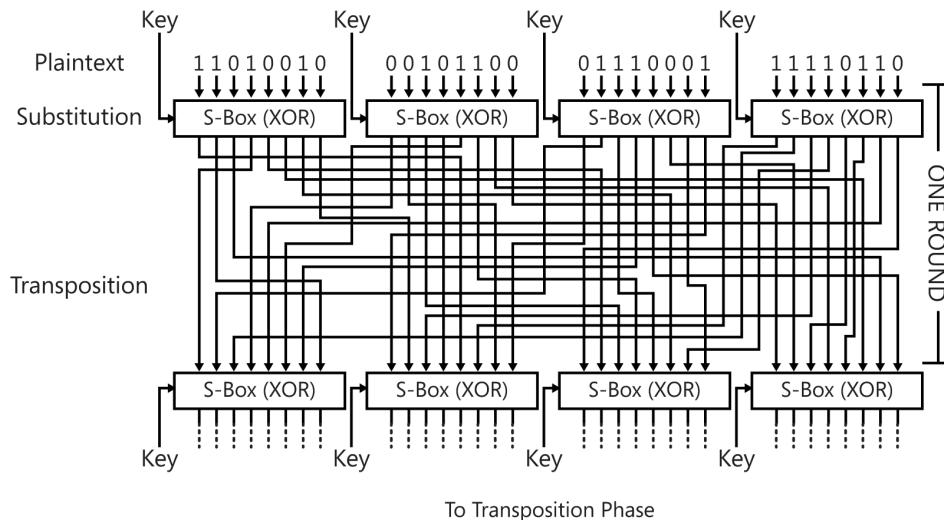


FIGURE 3-15 One round equaling one substitution (S-box) function and one transposition function

In general, symmetric key block ciphers code well in applications, and that code processes efficiently on CPUs, so it is said that *symmetric key block ciphers are best implemented in software*. This is compared to the implementation of symmetric keystream ciphers described earlier in this chapter.

Following is a list of symmetric key block ciphers to become familiar with:

- DES, 2DES, 3DES
- AES
- IDEA
- RC5, RC6
- Blowfish, Twofish
- CAST
- MARS
- SAFER



EXAM TIP

Several of these are examined in detail in this chapter. Learn them. Be aware of the others, not so much to know details about what they are, but as to know what they are not. They might be used as distractors on the exam. Just know that they are symmetric key block ciphers.

Data Encryption Algorithm (DEA) and Data Encryption Standard (DES)

DES was established as the US recommendation for protecting sensitive but unclassified content in 1976. DES is based on the Lucifer algorithm that was developed by Horst Feistel at IBM. Lucifer uses a 128-bit block and a 128-bit key, but it was redesigned as DEA to use a 64-bit block with a 56-bit key plus 8 parity bits to form 64 bits of key material. DES uses 16 rounds of substitution and transposition. In the late 1990s, DES was cracked and is now considered insecure.

Double DES (2DES)

In the late 1990s, when DES was cracked, researchers looked for a quick replacement, and 2DES was considered. However, after analysis, it was concluded that 2DES suffered from a vulnerability that allowed a known plaintext, meet-in-the-middle attack, which showed that 2DES was negligibly more secure than DES. For this reason, 2DES was and is considered insecure. An attacker who knew the plaintext and resulting ciphertext that was encrypted using double DES could brute-force the encryption process on the plaintext. The attacker would then brute-force the decryption process on the ciphertext. By comparing the brute force–encrypted messages with the brute force–decrypted messages and locating the two matching messages, the attacker would know the two keys used in the 2DES process.

Triple DES (TDES or 3DES)

In 1999, 3DES was added as the recommended implementation of the DES algorithm by the US government for protecting sensitive but unclassified content. 3DES passes plaintext through DES three times, using two or three keys. This increased the work factor back into a reasonable but still largely unsatisfactory level. The search began for the next generation of encryption algorithms.

3DES can operate in several modes, using two or three keys and performing the processes in different directions. The four modes of 3DES are:

- EEE-3 - Encrypt with key 1, then encrypt using key 2, then encrypt using key 3
- EEE-2 - Encrypt with key 1, then encrypt using key 2, then encrypt using key 1
- EDE-3 - Encrypt with key 1, then decrypt using key 2, then encrypt using key 3
- EDE-2 - Encrypt with key 1, then decrypt using key 2, then encrypt using key 1

These modes are shown in Figure 3-16.

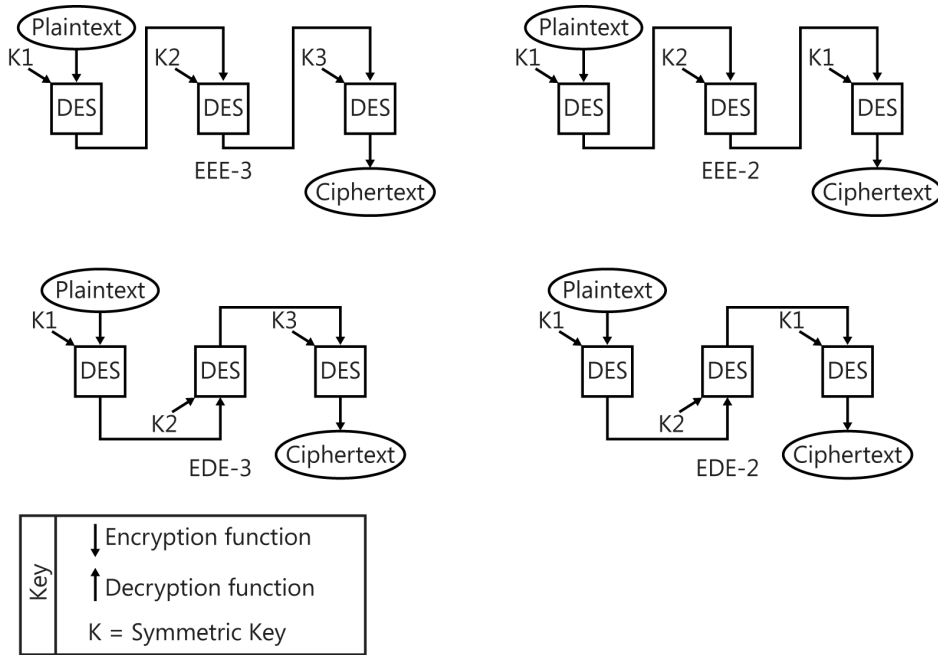


FIGURE 3-16 The modes of 3DES

The use of two keys is a tad more efficient than having to generate and securely distribute three keys, but it is also considered a tad less secure. 3DES is now considered insecure and, in 2002, was replaced with AES as the US recommendation for protecting sensitive but unclassified content.

Advanced Encryption Standard (AES)

AES is an iterative symmetric key block cipher based on the Rijndael algorithm, developed by Belgian cryptographers Vincent Rijmen and Joan Daemen. It uses a block size of 128 bits and can use key sizes of 128 bits, 192 bits, or 256 bits. It performs 10, 12, or 14 rounds of substitution and transposition, based on the key size used. It performs four basic crypto functions to accomplish confusion and diffusion:

- **AddRoundKey** An XOR substitution function
- **ShiftRows** A transposition function
- **SubBytes** A byte-level substitution (S-box) function
- **MixColumns** A mathematical function

On May 26, 2002, AES became the US recommendation for protecting sensitive but unclassified content, and it remains so currently. It is FIPS 197-compliant and is expected to perform in this role until approximately the year 2020.

The NSA has approved AES to protect Secret and Top Secret classified content.

It is said that if you could crack DES in one second, AES would take 149 trillion years to crack. The universe is only approximately 13.7 billion years old, so that work factor should be satisfactory for most purposes.

International Data Encryption Algorithm (IDEA)

IDEA was first described in 1991 and was intended to be the replacement for DEA/DES. It was developed under contract in Zurich by James Massey for a Dutch networking and telecom corporation. IDEA has been patented in numerous countries by MediaCrypt AG. IDEA is used in Pretty Good Privacy (PGP) and OpenPGP. It uses a 64-bit block size, a 128-bit key, and performs eight rounds of substitution and transposition.

A newer version of IDEA, called IDEA NXT, was released in 2005.

Rivest Cipher 5 (RC5) and RC6

Rivest Cipher v5 (RC5) was released in 1994. It can use block sizes of 32 bits, 64 bits, or 128 bits and key sizes ranging from 0 bits to 2,040 bits. Although the original implementation recommended 12 rounds, the algorithm supports 1 to 255 rounds. RC5 introduced data-dependent rotations and relatively simple implementation. RC5 is a proprietary algorithm and must be licensed from RSA for use. In response to a challenge by RSA, Distributed.net has cracked RC5 keys of 56 bits and 64 bits and is currently cracking the 72-bit keyspace by using a brute-force attack. At its current pace, the brute-force attack is expected to take 90 years to complete.

Rivest Cipher v6 (RC6) was proposed as a submission to replace DES and 3DES when AES won the competition for adoption as the NIST recommendation. RC6 was released in 1998. It uses a block size of 128 bits; key sizes of 128 bits, 192 bits, or 256 bits; and 20 rounds. RC6 uses data-dependent rotations, like RC5, and relatively simple implementation. Because of its submission as a candidate for a US standard, RSA was willing to provide free licensing for the use of RC6. However, although RSA has declared nothing certain, because RC6 was not adopted as the NIST standard, the company has kept its rights open to require licensing and royalty payments for its use.

Blowfish and Twofish

Designed in 1993 by Bruce Schneier, Blowfish has stood the test of time and remains popular in use in virtually every facet of computing and networking. There are currently no known successful attacks on Blowfish. The algorithm was released into the public domain and can be used freely. Blowfish uses a 64-bit block size and can use key sizes ranging from 1 bit to 448 bits. It performs 16 rounds of substitution and transposition.

A closely related algorithm by Schneier is Twofish, considered by many to be the second generation of Blowfish. Twofish was released in 1998 and has not (yet) been cracked. It uses a 128-bit block size and can use key sizes ranging from 128 bits to 256 bits. It performs 16 rounds of substitution and transposition. Twofish has also been released into the public domain and can be used freely.

Modes of symmetric key block ciphers

Remember that in the best situation, the bad guys only ever get to see ciphertext. They try to figure out the encryption key by analyzing the ciphertext and looking for patterns or other clues to the nature of the key. If the bad guys can figure out the key, they can steal all messages. Ciphertext that does not show these patterns and clues to the key is the result of a strong cryptosystem. Some cryptosystems are better at this than others are.

Generally speaking, the stronger the cryptosystem, the higher the price you will pay in performance and perhaps even for crypto-hardware (cost versus security). So when might you need to pay the higher price to get stronger security? Following are some examples:

- When you are protecting very sensitive (valuable) content
- When you are communicating in or through very hostile environments
- When you are producing large amounts of encrypted content
- When performance (cost) is less important than security (when cost is not an issue)

When you need to pay a higher price for stronger security from your cryptosystem, you have a range of choices. As described earlier, symmetric key block ciphers are quite fast (less expensive)—about 1,000 times faster than stream ciphers. However, the ciphertext from block ciphers tends to show patterns that might reveal the nature of the key used to produce the ciphertext. This, of course, is a bad thing. Cryptographic components can be added to the core block cipher within the cryptosystem to improve the strength of the ciphertext. These additional cryptographic components improve the strength by further separating the encryption key from the resulting ciphertext, further abstracting the encryption key from the resulting ciphertext, and further randomizing the ciphertext. There are five standard modes that symmetric key block ciphers can use that provide a range of performance versus strength to choose from. Each mode offers different combinations of cryptographic components or different orders of processing into the cryptosystem to randomize the ciphertext. The five modes of block ciphers are:

- Electronic Code Book (ECB)
- Cipher block chaining (CBC)
- Output Feedback mode (OFB)
- Cipher Feedback mode (CFB)
- Counter mode (CTR)

These five modes can be applied to any symmetric key block cipher, including DES, 3DES, IDEA, and AES. You will need to know these five modes and be generally familiar with how they operate.

Electronic Code Book (ECB)

Electronic Code Book (ECB) is the fastest and weakest mode of symmetric key block ciphers. It simply performs its encryption on each block of plaintext, as shown in Figure 3-17.

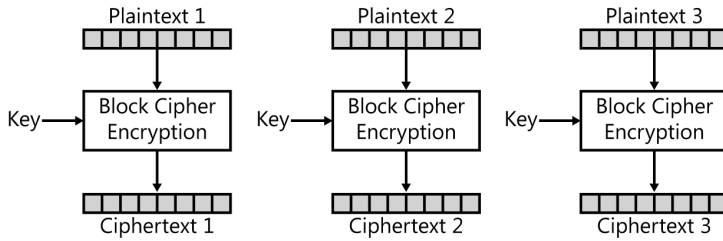


FIGURE 3-17 Electronic Code Book (ECB) mode encryption

There is no effort or attempt to randomize the ciphertext other than what happens within whatever block cipher was used. In theory, identical blocks of plaintext could produce identical blocks of ciphertext. This would be a very obvious pattern for the bad guys to see and guide them toward revealing the encryption key. Even without identical blocks of plaintext, patterns might be readily observed in ECB ciphertext.

ECB is typically used to protect small amounts of data, such as on personal identification numbers (PINs) on electronic payment terminals and ATMs, to keep the sample of ciphertext small. Smaller samples make it harder to recognize patterns that would present themselves readily in larger samples.

Cipher block chaining (CBC)

A little slower and a little stronger than ECB is cipher block chaining (CBC). This mode is fast enough and strong enough for many applications and has been the most popular mode of block ciphers. CBC was invented by the cryptography team at IBM in 1976 and adds one step to ECB. It applies an XOR with an IV to the first block of the plaintext (PT1). (This requires the IV to be the same size as the plaintext block.) Then it passes that through whatever symmetric key block cipher is being used to produce ciphertext block 1 (CT1). Then, CBC XORs a copy of CT1 with the next block of plaintext, PT2, *chaining the ciphertext* into the next block of plaintext. This block is passed through the block cipher to produce ciphertext block 2, CT2. A copy of CT2 is XORed with PT3, and so on, as shown in Figure 3-18.

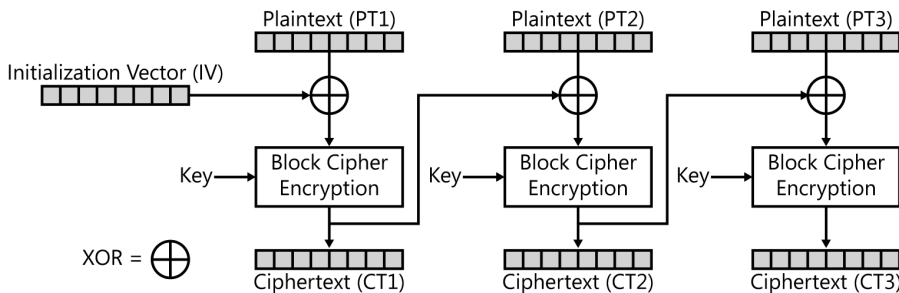


FIGURE 3-18 Cipher Block Chaining (CBC) mode encryption

This extra XOR step helps randomize the resulting ciphertext, reducing the likelihood of patterns that might lead to revealing the key. Notice that if any bit in the plaintext is altered, accidentally or intentionally, that change will propagate to and affect every block of ciphertext from that point forward to the end of the message.

CBC is slower than ECB because of the extra XOR processing and because the second block of plaintext cannot be processed until the first block of plaintext has been encrypted. (And the third block cannot be processed until the second block has been processed.) CBC must be processed sequentially rather than in parallel like ECB.

Output Feedback mode (OFB)

Output Feedback mode (OFB) is often called a stream cipher mode of a block cipher. OFB makes the block cipher behave like a stream cipher because it takes an IV as the first block of input where plaintext data would normally be entered, and it encrypts the IV by using the symmetric key. This behavior is similar to the stream cipher. The key and IV are inputs into the block cipher, as they are in a keystream generator. The output from this process is XORed with the first block of plaintext, PT1, to produce the first block of ciphertext, CT1, again as with a stream cipher. Next, the *output from the block cipher* (this is the original IV that has been encrypted—not the encrypted plaintext) is used as the IV input for the next block. This output feedback continues across all blocks until the entire message has been encrypted. The OFB process is presented in Figure 3-19.

OFB requires sequential processing, like CBC. OFB does not propagate errors in the plaintext through remaining blocks of ciphertext, whereas CBC does.

Further, the IV can be processed (encrypted) without the plaintext, so if there is any delay in accessing the plaintext—for example, due to reading it from a slow hard disk drive or waiting for the user to finish typing it in—much of the preprocessing can be accomplished and ready when the plaintext arrives. This output can also be fed to the next block for preprocessing. Then the relatively fast XOR process is the only thing left to complete to produce the ciphertext. This can improve the performance of OFB mode.

OFB is often used to protect satellite communications.

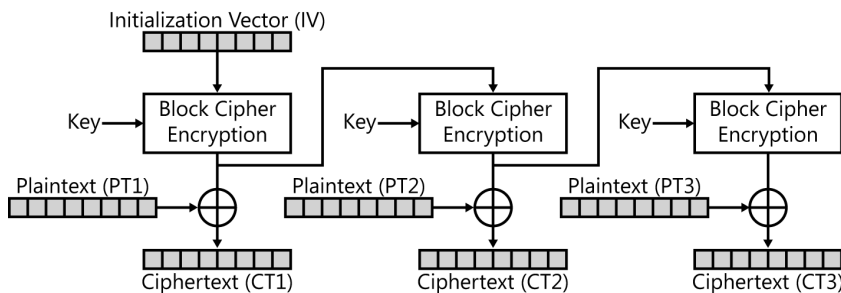


FIGURE 3-19 Output Feedback (OFB) mode encryption

Cipher Feedback mode (CFB)

Cipher Feedback mode (CFB) is another stream cipher mode of a block cipher. It operates like a cross between OFB and CBC. CFB encrypts an IV as the first block of input into the block cipher where plaintext data would normally be by using the symmetric key. The output from the block cipher is then XORed with the plaintext block 1 (PT1) to produce ciphertext block 1 (CT1). Then CFB takes a copy of CT1 to the next block cipher process and encrypts it by using the symmetric key, *feeding the ciphertext* into the block cipher. This CT1 block is passed through the block cipher, and the output is XORed with the second block of plaintext, PT2, to produce ciphertext block 2, CT2. A copy of CT2 is encrypted and then XORed with PT3, and so on, as shown in Figure 3-20. This ciphertext feedback continues across all blocks until the entire message has been encrypted.

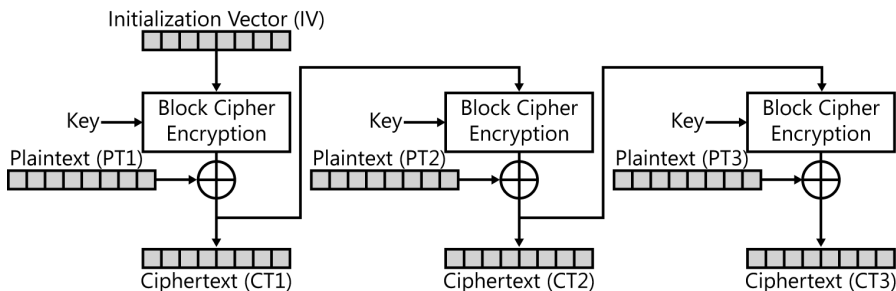


FIGURE 3-20 Cipher Feedback (CFB) mode encryption

CFB must be processed sequentially; note that if an error occurs during the encryption of the data in any block, that error will be propagated in each subsequent block through the rest of the message, just as in CBC mode. In contrast, when using OFB, errors in any one block of encryption will not propagate beyond that one block because the encrypted data is not chained.

As in OFB, the IV in CFB can be preprocessed (encrypted) without the plaintext. In CFB, however, the XOR function must be completed before the ciphertext output can be fed to the next block for processing.

CFB is often used to encrypt the mouse clicks and keystrokes upstream, and the video content downstream, within terminal services and RDP communications.

Counter mode (CTR)

Counter mode (CTR) is the newest mode of the group. CTR is both fast and strong. It adds another component to the process: a counter. This counter value is combined with an IV (also called a *nonce*) to produce the input in the symmetric key block cipher. This value is then encrypted through the block cipher by using the symmetric key. The encrypted output from the block cipher is then XORed with the first block of plaintext (PT1) to produce the first block of ciphertext, CT1. For the next block, the same IV is combined with the next value from the

counter and is encrypted using the symmetric key. This output is XORed with PT2 to produce CT2, and so on. Counter mode is presented in Figure 3-21.

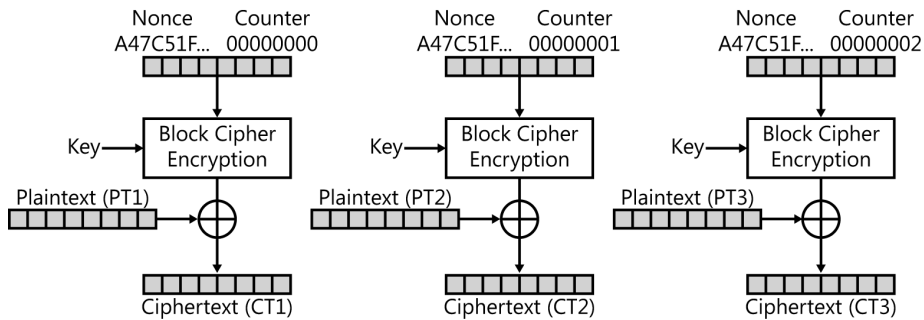


FIGURE 3-21 Counter (CTR) mode encryption

Notice that nothing is chained from one block to the next. This is how CTR mode gets its speed. Multiple blocks can be processed simultaneously, as in ECB. In addition, any errors in the plaintext will not propagate to subsequent blocks of ciphertext. In CTR mode, the nonce and counter can be preprocessed (encrypted) without the plaintext. This improves the performance of CTR mode. The extra cryptographic functions (when compared to ECB) used to abstract the key and produce stronger ciphertext make CTR quite strong.

The counter function can be as simple as a true counter, progressing from 0 to 1, 2, 3, and so on. However, many believe this is too simplistic. Remember that the IV (nonce) is a nonsecret value, so the bad guy might see it being sent from sender to recipient. And if the counter value is easily guessed and predictable, following a pattern, this value might also be easily obtained by the bad guys. It is these two values that are actually encrypted using the block cipher. Therefore, the bad guys could actually know the plaintext (the IV and the counter value) that goes into the block cipher. This is referred to as the *known plaintext attack* and provides the bad guys with more information than many are comfortable with. They feel that this is too much information and therefore choose a much more complex counter function with substantially less predictability.

CTR is used in the strongest implementation of 802.11i, W-Fi Protected Access v2 (WPA2), and is used in IPsec VPNs.

NOTE KNOWN PLAINTEXT ATTACK?

There will be more on the types of attacks on cryptography in the “Attacks on cryptography” section later in this chapter.

Signing and sealing using symmetric key algorithms

Although IT professionals know that there are five desirable cryptographic services, from a user's perspective, signing and sealing are the two recognized services needed within information systems. They are often the check boxes available in an application.

- Signing messages provides authentication and integrity validation services.
- Sealing provides confidentiality services.

Symmetric key cryptosystems can provide these services, but in a limited sense. As you recall from earlier in this chapter, the following list of services shows what symmetric key cryptography can provide and at what level of strength:

- **Confidentiality** Strong
- **Authentication** Weak
- **Nonrepudiation** No
- **Integrity** Weak
- **Secure key distribution** No, the use of symmetric keys causes the problem

If you receive a message that decrypts correctly when you use a key you share only with the user BoBo, you have a pretty good idea that the message really came from BoBo. Nevertheless, remember that because there are at least two copies of the symmetric key, it is not possible to prove a single and undeniable source. There is a potential that the other key was used to produce the message, and it is never certain or provable which copy of the key was actually used to produce the ciphertext. This limitation provides only a weak level of authentication and does not provide nonrepudiation. This weaker form of authentication based on symmetric keys is often called *data origin* or *system authentication*.

If a hashing function is added to the symmetric key cryptosystem, it can be used to prove that a message has not been tampered with since it was sent, but if you cannot prove who sent the message, the integrity (level of trust) for the message cannot be any greater than the level of trust for the authentication. However, it is an indication of some level of trust for the integrity of a message, and that is better than no indication of integrity.

Signing by using symmetric key algorithms

When symmetric keys are used to provide authentication and integrity validation of messages, it is referred to as *message authentication code (MAC)*. When this term is used, it is understood that the authentication and integrity validation is weak but better than nothing. Several techniques can provide MACs:

- Hashed message authentication code (HMAC)
- Cipher block chaining message authentication code (CBC-MAC)
- Cipher-based message authentication code (CMAC)

NOTE WHICH MAC IS IT?

When it comes to cryptography and hash values, the MAC acronym represents message authentication code. However, when it comes to endpoints and a frame on a network, MAC refers to the Media Access Control address used to identify each node or hop in the transmission of the frame. With access control models, MAC refers to the mandatory access control model used primarily by governments and the military because of its strength. Another use of this very popular acronym is that of MAC times used in digital forensics, which deal with the time stamps recorded by most file systems that show when a file or directory has last been Modified (written to), Accessed (read from), and Created (first recorded on the volume). When you see the term MAC, think for a moment what your topic is and then be sure you are thinking of the correct MAC.

MAC VERSUS DIGITAL SIGNATURE

Although the use of a MAC on a message might be called “signing the message,” MAC uses symmetric keys and provides only weak authentication and weak integrity validation. It should not be confused with signing a message by using a digital signature (covered in the “Cryptography in use” section later in this chapter). Digital signatures use asymmetric keys and (typically) public key infrastructure (PKI) digital certificates along with a hashing algorithm. Digital signatures provide strong authentication, strong nonrepudiation, and strong integrity validation.

MAC and digital signatures are competing technologies; MAC is regarded as the poor man’s version (faster and cheaper but weaker) of a digital signature.

MAC types of authentication are used in the open standard Challenge Handshake Authentication Protocol (CHAP) and Microsoft proprietary implementation of CHAP, MS-CHAP, currently in version 2. These are forms of a zero-knowledge proof, in which a user can prove his identify (in a weak sense) without revealing the symmetric key (the user’s password) to the authentication service. CHAP is used in Kerberos protocol, a symmetric key authentication system. CHAP, Kerberos protocol, and the zero-knowledge proof are covered in Chapter 2, “Access control.”

HASHED MESSAGE AUTHENTICATION CODE (HMAC)



Hashed message authentication code (HMAC) is performed by adding a symmetric key to a message and then running the message and key through a hashing algorithm. This produces a MAC value, called the sender’s MAC or MACs in Figure 3-22, which is the sender’s hash value. (Actually, it is more than a hash value because it contains properties of the symmetric key as well as properties of the message.) Next, the plaintext message (without the symmetric key) and the MAC value are sent to the recipient. The recipient would have to have acquired a copy of the symmetric key through some other secure mechanism, as always.

To verify the HMAC, the recipient adds her copy of the symmetric key to the message and runs the message and key through the same hashing algorithm. This produces a MAC value, called the recipient’s MAC or MACr in Figure 3-22, which is the recipient’s MAC, the hash

value that includes the recipient's copy of the symmetric key. If the message has not been modified, and if the sender and recipient have the correct and same copies of the symmetric key, MACs should equal MAC_r , as shown in Figure 3-22. In this case, the authenticity of the sender and the integrity of the message can be trusted to some level but cannot be proven.

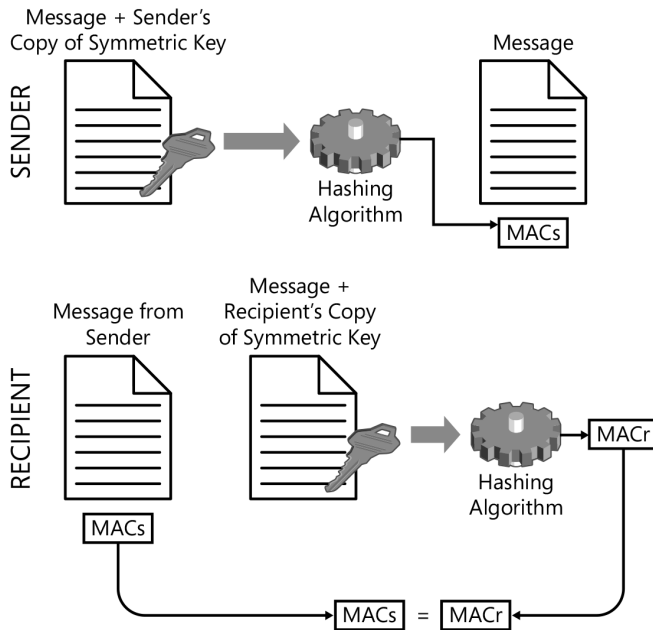


FIGURE 3-22 Hashed message authentication code (HMAC)

If the two MAC values are not equal, either the message *has* been modified, or the sender and recipient *do not* have the correct and same copies of the symmetric key. The message cannot be trusted, and most applications will discard the message.

Notice that no encryption was involved, and no confidentiality was provided. That is not the objective of signing a message. If you need confidentiality, you need to seal the message, not sign it. HMAC only provides weak authentication and weak integrity validation.

CIPHER BLOCK CHAINING MESSAGE AUTHENTICATION CODE (CBC-MAC)



Cipher block chaining message authentication code (CBC-MAC) operates quite differently. Remember the CBC discussion in an earlier section of this chapter—understand the statement that errors or alterations in the plaintext propagate through all ciphertext blocks, following the error to the end of the message. This is the basis for CBC-MAC.

The sender runs the message through a block cipher in CBC mode, using her copy of the symmetric key. Each block of ciphertext is XORed with the next block of plaintext. This is what carries any alterations forward through the message. The very last block of the sender's CBC ciphertext, S in Figure 3-23, is the culmination of every bit contained in the message and is a product of the sender's symmetric encryption key.

The plaintext message and the very last block of CBC ciphertext are sent to the recipient. The recipient verifies the signature by running the plaintext message through the same block cipher in CBC mode and using the recipient's copy of the symmetric key. The very last block of the recipient's CBC ciphertext, R in Figure 3-23, is then compared to the block of CBC ciphertext from the sender, S. If R and S are equal, then the message can be trusted, to some level, as not modified. It can be trusted on some level that the message was most likely sent by the claimed sender because only the sender and recipient share the correct and same copies of the symmetric keys. In this case, the authenticity of the sender and the integrity of the message can be trusted to some level but cannot be proven.

CBC-MAC is shown in Figure 3-23.

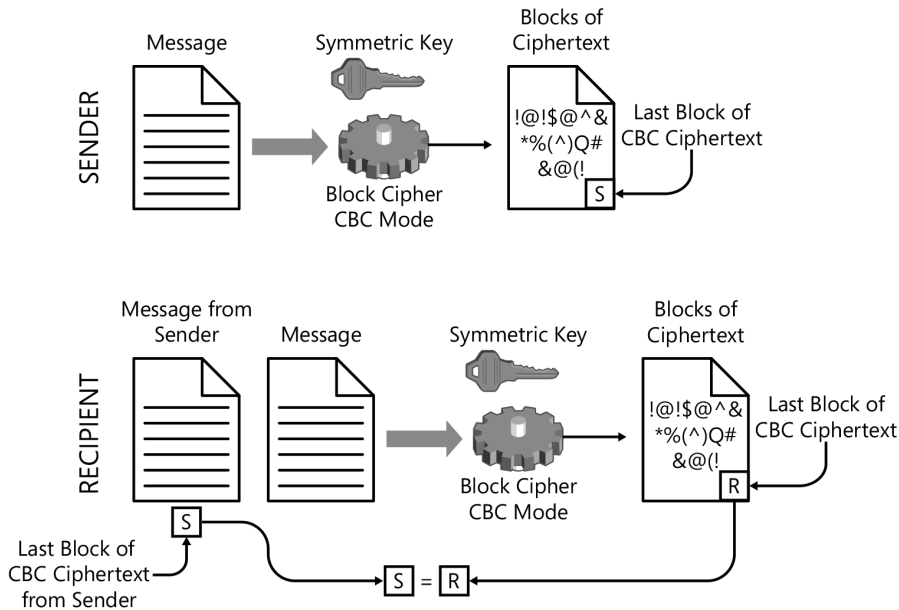


FIGURE 3-23 Cipher block chaining message authentication code (CBC-MAC)

If the two CBC-MAC values are not equal, either the message *has* been modified, or the sender and recipient *do not* have the correct and same copies of the symmetric key. The message cannot be trusted, and most applications will discard the message.

Notice that encryption *was* involved, but no confidentiality was provided. The message was sent to the recipient in plaintext. Confidentiality is not the objective of signing a message. If you need confidentiality, you need to seal the message, not sign it. You could sign and seal a message, but that is two different functions and two different security objectives. CBC-MAC provides only weak authentication and weak integrity validation.

CBC-MAC had been approved in FIPS Publication 113, using DES as the block cipher, and is included in the International Organization for Standardization/International Electromechanical

Commission ISO/IEC 9797-1 (1999) but is now considered insecure for variable-length messages. CBC-MAC is used in the strongest implementation of 802.11i WPA2.

CIPHER-BASED MESSAGE AUTHENTICATION CODE (CMAC)

Cipher-based MAC (CMAC) is an improved variant of CBC-MAC that was developed to correct the vulnerabilities in CBC-MAC on variable-length messages. CMAC is documented in the NIST Special Publication series 800-38B, May 2005, which is the US recommendation for performing authentication with symmetric key block ciphers.

Sealing by using symmetric key algorithms

Symmetric keys are commonly used to provide confidentiality. They can provide strong confidentiality and strong encryption. When cryptography is used to provide confidentiality, it is often called sealing the message, like putting the message in an envelope and sealing (encrypting) it in a way that only the intended recipient can open (decrypt) the envelope.

Weaknesses in symmetric key algorithms

Most of these issues have been mentioned already, but the topic is so broad that a summary is warranted.

Basic substitution ciphers (confusion) are easily cracked by using frequency analysis. Transposition ciphers (diffusion) contain all the characters in the original plaintext message and simply need to be correctly repositioned. Used alone, each of these is pretty easy to crack, but when the two are used together, they can produce strong ciphertext.

If two different keys produce the same ciphertext from the same message, this is called *key clustering*. Key clustering is an undesirable characteristic of some cryptosystems because it means that two (or possibly more) different keys could also decrypt the secure content. This increases the chances for attackers to guess the key and statistically reduces the time it will take to crack any one of the keys that will decrypt the protected message and reveal the meaning of the message. A strong cryptosystem has a low frequency of key clustering occurrences.

Symmetric key cryptography only provides strong confidentiality services, and provides weak authentication and weak integrity verification services. The use of symmetric keys cannot provide non-repudiation, and they cause the problem of needing some mechanism to securely distribute the symmetric keys to the participants of the cryptosystem.

The next two weaknesses of symmetric key cryptosystems fall into the category of key management. First, somehow the sender must securely communicate the symmetric key to the recipient (secure key distribution), but if there were a way to accomplish this within the symmetric key cryptosystem, why not just tell the recipient the message securely? This problem historically was solved by establishing and sharing a symmetric key before it was needed usually when the sender and recipient were in close proximity in a nonhostile environment.

Finally, recall that the number of keys required within a symmetric key cryptosystem grows very rapidly as the number of users grows $(N \times (N - 1))/2$ and that typically two copies of each key must be protected for the lifetime of the cryptosystem.



EXAM TIP

Be sure you understand the strengths, weaknesses, and solutions available within symmetric key cryptosystems.

✓ Quick check

1. Why are symmetric key block ciphers weaker than symmetric keystream ciphers?
2. What new mode of block ciphers is both fast and strong?

Quick check answers

1. Block ciphers tend to show patterns in the ciphertext that might reveal clues to the nature of the encryption key.
2. Counter mode

Asymmetric key algorithms and cryptosystems

For nearly 5,000 years, only symmetric key cryptography existed, with all the weaknesses and problems that symmetric key cryptography introduced. The most critical of those issues was that of secure key distribution. How do you get a copy of the symmetric key to the recipient securely? In the 1940s, John Mauchly and John Eckert developed one of the first computers, the Electrical Numerical Integrator And Computer (ENIAC, 1947). In the 1950s and 1960s, computers and the computational power they bring became more available, allowing researchers and scientists to write programs to crunch larger and larger numbers. Then, in the early 1970s, with computers more available than at any time in history, the paths of two mathematician/cryptographers, Whitfield Diffie and Martin Hellman, intersected. They had both independently been working on a solution to this 5,000-year-old problem of secure key distribution. When they compared their research and began collaborating, they realized that together they had the solution.

The very first asymmetric key algorithm was introduced in 1976 to solve the problem of secure symmetric key distribution. It became known as the Diffie-Hellman algorithm (DH), a key agreement protocol. In 2002, Whitfield Diffie suggested adding the name of their co-inventor, Ralph Merkle, to the name of the algorithm, so you might sometimes see the name of the algorithm as Diffie-Hellman-Merkle.

This asymmetric algorithm not only provided the solution to secure symmetric key distribution, it cracked open a new era in cryptography. When other new asymmetric algorithms are used, all five of the desirable cryptographic services can be provided in a strong manner. Once again, they are:

- Confidentiality
- Authentication

- Nonrepudiation
- Integrity
- Secure key distribution



Asymmetric key cryptography uses two different but mathematically related keys. This key pair includes a *public key* and a *private key*. What one key encrypts, only the other key can decrypt.

- If you encrypt content by using your private key, the private key cannot decrypt the content. Only the public key can decrypt this ciphertext.
- If you encrypt content by using your public key, the public key cannot decrypt the content. Only the private key can decrypt this ciphertext.

The public key can be shared with anyone: a good guy, a bad guy, and even an unknown guy. Another person can do you no harm by having your public key.

The private key should never be shared or exposed to anyone. If the private key is exposed to someone else, this key pair should immediately be replaced and never used again. All content protected by the exposed key pair should be protected by using the replacement key pair.

With a public key, you can:

- Encrypt content (providing confidentiality, sealing) to the owner of the mathematically related private key.
- Verify (decrypt) the digital signature of the sender (signer) of a message.

With a private key, you can:

- Produce a digital signature by encrypting a hash value of a message.
- Decrypt content that has been encrypted with the mathematically related public key.



EXAM TIP

The relationship between the private key and the public key is a critical concept to understand in asymmetric key algorithms and cryptosystems. The functions that each provides are also critical knowledge for the exam.

Although symmetric algorithms are largely based on substitution (confusion) and transposition (diffusion) of binary bits, asymmetric algorithms are based primarily on mathematic formulas. The keys (numbers) used in asymmetric key cryptography are notably longer than those used in symmetric key cryptography. A short key in asymmetric key cryptography is 768 bits long (DH Group 1), providing a keyspace range of zero to approximately the number 15 with 230 zeros behind it. Today, this key size is considered too short to use. In 2003, the IETF published RFC 3526, which identifies key lengths up to 8,192 bits (8 kilobits, known as DH Group 18), providing a keyspace range of zero to approximately the number 1 with 2,466 zeros behind it. These keys are so large and unique (random) that it is statistically impossible for someone to guess your private key. The recommendation today is to use a key longer than 1 kilobit, and typically 2 kilobits or 3 kilobits in length is preferred, with the NIST recommendation in 2011 to use a 2-kilobit key minimum.

Key management in an asymmetric key cryptosystem is much easier than in a symmetric key cryptosystem. Only two keys are required per user ($2N$, where N is the number of users) in the asymmetric key cryptosystem. Further, only the private key requires protection, cutting in half this very low number of $2N$ keys. Because you can readily share the public key with anyone, there is no problem of secure key distribution, and the asymmetric key cryptosystem solves this problem for the symmetric key cryptosystems. Secure key distribution will be described in full detail in the “Cryptography in use” section later in this chapter.

The primary detriment with asymmetric key cryptosystems is the performance. Crunching numbers that fill a page takes some processing power and time. With this in mind, cryptographers have assembled hybrid cryptosystems that take advantage of the best features of the various cryptographic functions.

The great security benefits provided by the asymmetric key cryptosystem has led to its use with digital certificates and the PKI system that has become a standard for security in contemporary information systems. Another hybrid cryptosystem of note is Pretty Good Privacy (PGP). These two commonly used technologies are explored in more detail in the “Cryptography in use” section later in this chapter.

Signing by using asymmetric key algorithms in a hybrid cryptosystem



When asymmetric keys are used to provide authentication and integrity validation, it is called a *digital signature*. Because of the strength of the authentication provided by the asymmetric key pair, the digital signature also provides nonrepudiation. This digital signature uses a hashing function and an asymmetric key pair. It also typically uses *digital certificates*, provided by a PKI and *certification authorities (CAs)*, the systems that produce the digital certificates. The digital certificate adds the element of trust to the cryptographic functionality of the asymmetric key pair. The digital certificate binds the user’s identity and the user’s asymmetric public key to the X.509 digital certificate and is then signed by the CA that created the certificate. PKI is described in the “Cryptography in use” section later in this chapter, but for now, it is enough for you to know that after a recipient verifies a digital certificate, the recipient knows and believes that the public key embedded in the certificate belongs to the user whose name is documented on the certificate.

The digital signature provides the following three cryptographic services:

- **Authentication** Strong
- **Nonrepudiation** Strong
- **Integrity validation** Strong

If a sender digitally signs a message, he proves these three things to the recipient. The digital signature does not provide confidentiality services. If you need confidentiality, you need to seal the message, not sign it.

Here is how the digital signature (signing), using asymmetric keys and a hashing function, works. Suppose that BoBo (the sender) wants to send LuLu (the recipient) a digitally signed message to prove to LuLu that he sent the message, that he cannot deny sending the message, and that the message has not been altered since BoBo created the message.

1. BoBo, the sender, has an asymmetric key pair. The public key is embedded in a digital certificate from a trusted CA. BoBo holds the mathematically related private key securely and never shares the private key with anyone.
2. BoBo creates a message.
3. BoBo runs a hashing algorithm on the entire message, producing a hash value as the output of the hashing algorithm. The hash value acts as a fingerprint of the message. If any one or more binary bits that make up the message are changed, the hash value will change. Because BoBo, the creator of the message, is running the hashing algorithm on the message at the time of creation, the message is known good at this time.
4. BoBo uses his private key to encrypt the hash value of the message. This is the digital signature for the message. BoBo is the only person on the planet who can access this private key, therefore BoBo is the only person on the planet who can produce this signature for the message.
5. BoBo sends the plaintext message and the digital signature (the encrypted known good hash value) to LuLu, the recipient (see Figure 3-24).

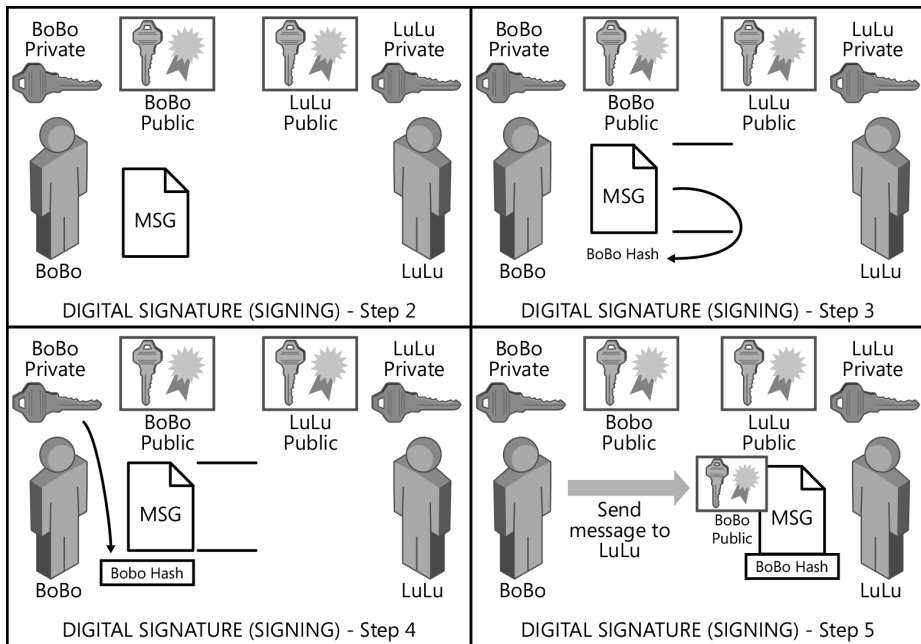


FIGURE 3-24 Producing the digital signature

6. Upon receiving the digitally signed message, LuLu acquires BoBo's digital certificate.
7. LuLu verifies BoBo's digital certificate and concludes, with certainty, that the public key embedded in the BoBo digital certificate belongs to BoBo.
8. LuLu uses BoBo's public key from the BoBo digital certificate to decrypt the digital signature and reveal the hash value that BoBo created at the time when the message was known good.
9. LuLu runs the same hashing algorithm on the entire message and produces a second hash value, the LuLu at-time-of-use hash value.
10. LuLu compares the BoBo known good hash value to the LuLu at-time-of-use hash value (see Figure 3-25).

If the two hash values are identical, LuLu can conclude that three cryptographic services have been provided strongly.

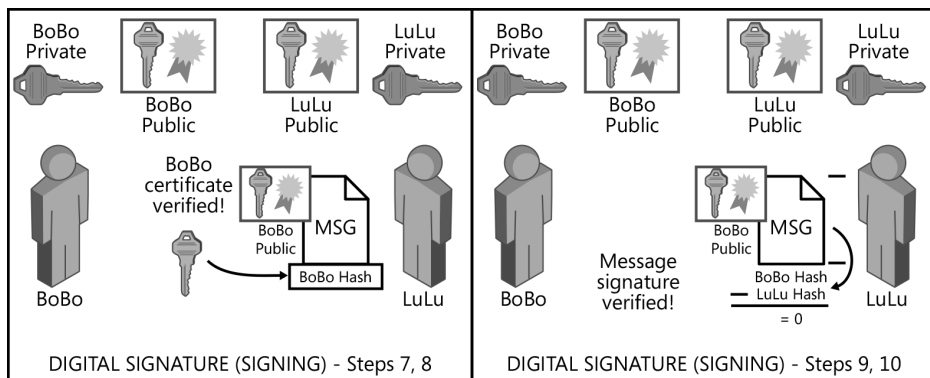


FIGURE 3-25 Verifying the digital signature

11. Because the two hash values are identical, LuLu knows that the message has not been modified from the time it was known good by BoBo, at the time of creation. This is *strong integrity validation*.
12. Because it was BoBo's public key that decrypted the digital signature correctly (so that the hash value was correct), LuLu knows that only BoBo could have encrypted the hash value by using the mathematically related BoBo private key. This is *strong authentication*.
13. Because of the strong authentication services provided by asymmetric key cryptography, LuLu knows that BoBo cannot deny being the sender of the message. He is the only one on the planet who can access the BoBo private key that could have encrypted the digital signature such that the mathematically related BoBo public key decrypts it correctly. This is *strong nonrepudiation*.

Sealing by using asymmetric key algorithms in a hybrid cryptosystem

Sealing messages means providing confidentiality for messages. Although asymmetric key cryptography can provide strong confidentiality, remember that it is approximately 1,000 times slower than symmetric key cryptography, and symmetric key cryptography can provide strong confidentiality. The messages sent between users in an information system could be large—perhaps tens of megabytes, hundreds of megabytes, or even several gigabytes. You wouldn't want to encrypt that bulk with asymmetric keys that are 1,000 times slower when the much faster symmetric keys can provide good, strong ciphertext. The symmetric keys, however, are quite small, ranging from 64 bits to 512 bits in length. These small symmetric keys could easily be encrypted with the slow asymmetric keys to provide the secure key distribution that is required of symmetric key cryptosystems.



With this in mind, it becomes obvious why hybrid cryptosystems use *symmetric session keys* and asymmetric keys together to provide confidentiality (sealing) of messages. Here is how providing confidentiality of messages, with the use of asymmetric and symmetric keys, works:

1. LuLu, the recipient, has an asymmetric key pair. The public key is embedded in a digital certificate from a trusted CA. LuLu holds the mathematically related private key securely and never shares the private key with anyone.
2. BoBo, the sender, creates a message.
3. BoBo acquires LuLu's (the recipient's) digital certificate.
4. BoBo verifies LuLu's digital certificate and concludes, with certainty, that the public key embedded in the LuLu digital certificate belongs to LuLu.
5. BoBo uses a cryptographic service provider (CSP) on his local computer to create a pair of symmetric session keys. Remember that symmetric key cryptography uses small keys (typically only 64 bits to 512 bits) and is approximately 1,000 times faster than asymmetric key cryptography.
6. BoBo uses one of the symmetric session keys to encrypt the message intended for LuLu.
7. BoBo uses LuLu's public key from the LuLu digital certificate to encrypt the second copy of the symmetric session key. Remember that this copy of the symmetric session key can be used to decrypt the sealed message.
8. BoBo sends the encrypted message and the encrypted symmetric session key to LuLu (see Figure 3-26).

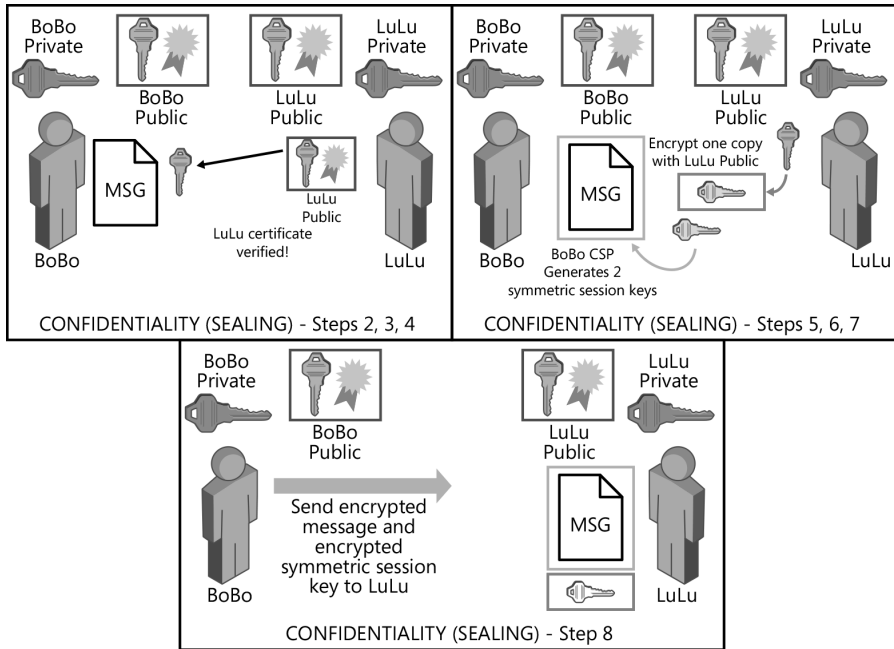


FIGURE 3-26 Sealing a message

9. LuLu receives the encrypted message that has been encrypted with the symmetric session key and the symmetric session key that has been encrypted with her public key.
10. Only LuLu has the private key that can correctly decrypt the symmetric session key. LuLu uses her private key to decrypt the symmetric session key.
11. LuLu uses the symmetric session key to decrypt the encrypted message, revealing the meaning of the message. The message was securely delivered to LuLu, and only LuLu could decrypt the message successfully. Secure key distribution (the symmetric session key) was also accomplished to facilitate the secure delivery of the message (see Figure 3-27).

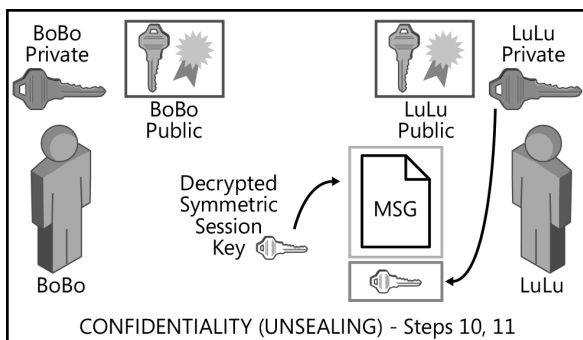


FIGURE 3-27 Unsealing a message

Sealing messages provides the cryptographic service of confidentiality and secure key distribution. Notice that the symmetric session key is used for one message only and is then destroyed after its one-time use, “session” implying short-term or temporary. Notice also that the public and private key pairs are long-term keys (also called static, persistent, or secret keys). These keys are typically good for one or two years before expiring.

Sending to multiple recipients when sealing

If the sender wants to send a message securely to multiple recipients, all the sender has to do is perform the following steps. See Figure 3-28.

1. Acquire and validate the digital certificate for each intended recipient.
2. Duplicate the symmetric session key, one copy for each intended recipient plus the one copy the sender needs to encrypt the message.
3. Extract the public key from the verified digital certificate for each intended recipient.
4. Encrypt a copy of the symmetric session key, one key at a time, using the public key of each intended recipient so that there is one encrypted symmetric session key for each intended recipient that has been encrypted using that recipient’s public key.
5. Distribute a copy of the encrypted message to each intended recipient along with a copy of the symmetric session key that has been encrypted with that recipient’s public key.

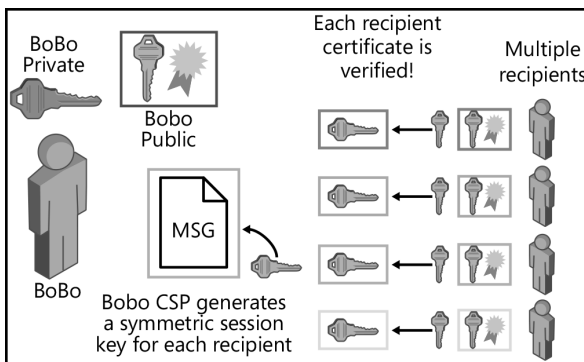


FIGURE 3-28 Sending a confidential message to multiple recipients

The intended recipient decrypts the copy of the symmetric session key by using her private key and then uses the decrypted symmetric session key to decrypt the securely distributed message.

Signing and sealing messages

So how would signing and sealing a message work? This would require all five desirable cryptographic services. Here is how the digital signature (signing) and sending of secure messages (sealing) works, using asymmetric keys, symmetric keys, and a hashing function: BoBo (the sender) wants to send LuLu (the recipient) a secure and digitally signed message. This keeps the message secret, for only BoBo and LuLu, and proves to LuLu that Bobo sent the message, that he cannot deny sending the message, and that the message has not been altered since BoBo created the message. The following steps detail the procedure (see Figures 3-29 and 3-30). Feel free to review the individual pieces as shown in Figures 3-24, 3-25, 3-26, and 3-27.

1. BoBo, the sender, has an asymmetric key pair. The public key is embedded in a digital certificate from a trusted CA. BoBo holds the mathematically related private key securely and never shares the private key with anyone.
2. LuLu, the recipient, has an asymmetric key pair. The public key is embedded in a digital certificate from a trusted CA. LuLu holds the mathematically related private key securely and never shares the private key with anyone.
3. BoBo creates a message.
4. BoBo runs a hashing algorithm on the entire message, producing a known good hash value as the output of the hashing algorithm.
5. BoBo uses his private key to encrypt the hash value of the message. This is the digital signature for the message.
6. BoBo acquires and validates a copy of the LuLu digital certificate, and concludes, with certainty, that the public key embedded in the LuLu digital certificate belongs to LuLu.
7. BoBo uses a cryptographic service provider (CSP) on his local computer to create a pair of symmetric session keys.
8. BoBo uses one of the symmetric session keys to encrypt the message intended securely for LuLu.
9. BoBo uses LuLu's public key from the LuLu digital certificate to encrypt the second copy of the symmetric session key. Remember that this copy of the symmetric session key can be used to decrypt the sealed message.
10. BoBo sends the encrypted message, the encrypted hash value of the message (the digital signature), and the encrypted symmetric session key to LuLu.

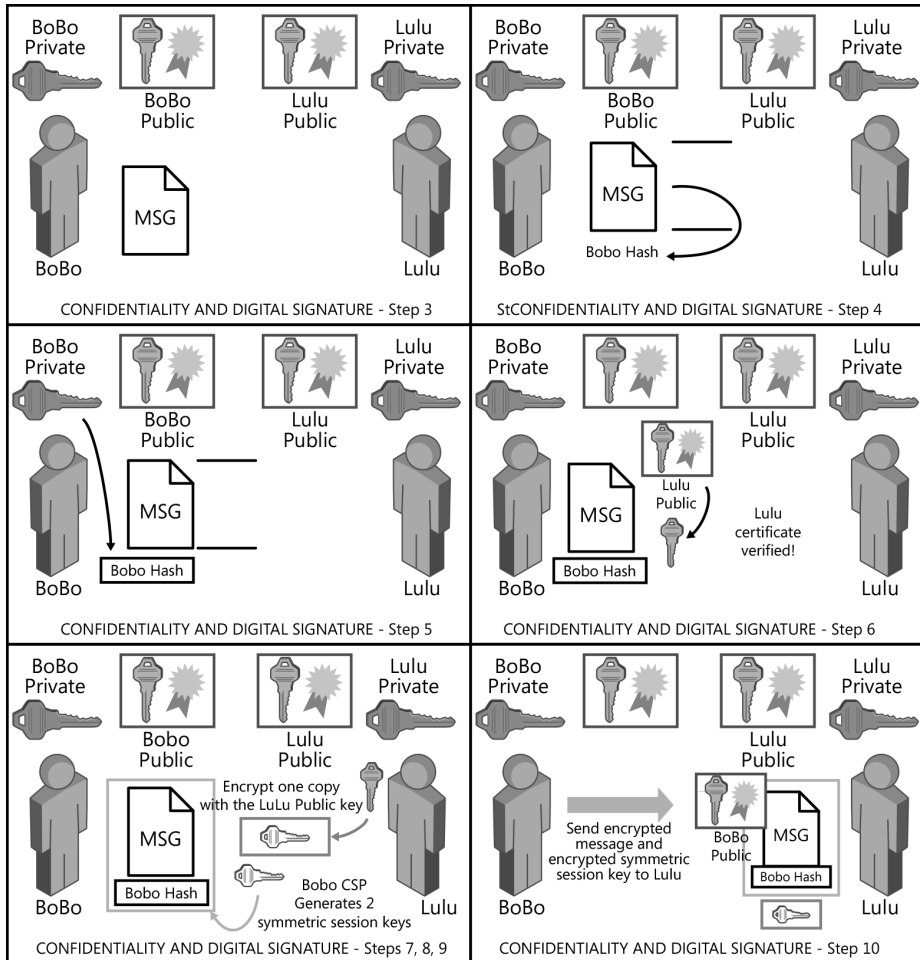


FIGURE 3-29 Signing and Sealing a message

- 11.** LuLu receives the encrypted message, the encrypted hash value of the message (the digital signature), and the encrypted symmetric session key that has been encrypted with her public key.
- 12.** LuLu uses her private key to decrypt the symmetric session key.
- 13.** LuLu uses the symmetric session key to decrypt the encrypted message, revealing the meaning of the message. The message was securely delivered to LuLu, and only LuLu could decrypt the message successfully. Secure key distribution (the symmetric session key) was also accomplished to facilitate the secure delivery of the message.

14. To validate BoBo's digital signature, LuLu acquires BoBo's digital certificate.
15. LuLu verifies BoBo's digital certificate and concludes, with certainty, that the public key embedded in the BoBo digital certificate belongs to BoBo.
16. LuLu uses BoBo's public key from the BoBo digital certificate to decrypt the digital signature and reveal the hash value that BoBo created at the time when the message was known good.
17. LuLu runs the same hashing algorithm on the entire message and produces a second hash value, the LuLu at-time-of-use hash value.
18. LuLu compares the BoBo known good hash value to the LuLu at-time-of-use hash value.
19. If the two hash values are identical, LuLu can conclude that three cryptographic services have been provided strongly: authentication, nonrepudiation, and integrity validation.

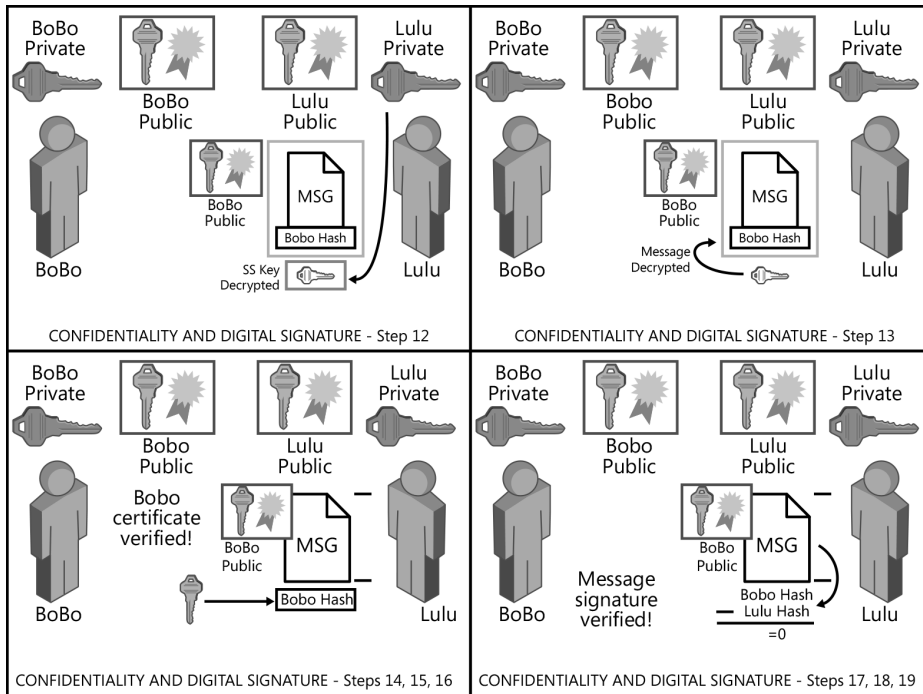


FIGURE 3-30 Unsealing a message and verifying the digital signature

Wow! If you were able to follow that, you might have become a cryptographer! (Did that happen again?!)

The signing and sealing of messages requires all five desirable cryptographic services and requires both BoBo, the sender, and LuLu, the recipient, to have an asymmetric key pair and (typically) digital certificates. Content can be signed and sealed and then sent to one or more recipients (protection for data in transit), or it can be signed and sealed to self, so that the creator can protect and verify the integrity of his own secret content in secure storage (protection for data at rest).

Asymmetric key algorithms

Asymmetric key algorithms have only been in existence since the mid-1970s, when computers became available to the masses, and they are based on relatively heavy mathematics with large, perhaps unbelievably huge, numbers.

Following is a list of the most commonly used asymmetric algorithms:

- Diffie-Hellman-Merkle (or just Diffie-Hellman [DH])
- RSA (Rivest, Shamir, Adleman)
- Elliptic Curve Cryptography (ECC)
- ElGamal
- Digital Signature Standard (DSS)
- LUC (Reference only)
- XTR (Reference only)

By culling data from RSA, NIST, IETF, and other sources, Table 3-5 shows how the key length used in different algorithms align with approximately equivalent strength.

TABLE 3-5 Comparison of approximately equivalent key strength vs. key length (values are estimates and shown in bits)

Symmetric	ECC	XTR	LUC	RSA
80	163	342	512	1,024
112	233	683	1,024	2,048
128	283	1,024	1,536	3,072
192	409	2,560	3,840	7,680
256	571	5,120	7,680	15,360



EXAM TIP

For the exam, be sure that you understand the mathematic function that each algorithm is based on and the services the specific algorithms or cryptosystems can provide. If you can remember more about each one, it might help, but at least remember the math behind the algorithm and the services each provides.

Invented in 1976 and based on large numbers and mathematics, asymmetric key cryptosystems are the latest and greatest addition to cryptography. They rely on a pair of mathematically related numbers (keys); one is a private key and the other is a public key. What one key encrypts only the other key can decrypt. You can share your public key with anyone. You can never share your private key with anyone. Because you have only the public key of another, you have no insight into the nature of the other's private key.

Most asymmetric key algorithms use variations of mathematics from the following list:

- Calculating discrete logarithms in a finite field
- Factoring large numbers into their prime values
- Calculating discrete logarithms in a finite field, limiting that finite field to pairs of numbers plotted on an elliptic curve or other mathematical or numeric sequences

Although some asymmetric key cryptosystems provide only a few services, when used with symmetric key algorithms and hashing algorithms (a hybrid cryptosystem), most can provide all five desirable cryptographic services in a strong manner. They form the basis of the PKI that adds the element of trust to the functionality of the asymmetric key pair. The PKI binds a user's public key to her identity by using an X.509 digital certificate and ensures trust through a hierarchical trust model.

The key lengths in asymmetric key cryptosystems are typically at least twice as long as symmetric keys and are often more than 10 times the length of symmetric keys. This causes them to be notably slower than symmetric key cryptosystems. The quantity of keys generated ($2n$) and protected (n) are substantially lower than those of symmetric key cryptosystems ($n \times (n-1)/2$), especially as the number of users participating in the cryptosystem grows.

Diffie-Hellman-Merkle (or just Diffie-Hellman)

Diffie-Hellman was introduced in 1976. It was designed to solve that 5,000-year-old problem with symmetric key cryptography—the necessity to distribute copies of symmetric keys securely without some prior relationship or arrangements. Following are several key concepts to understand about the Diffie-Hellman (DH) algorithm.

- DH was the first asymmetric algorithm ever.
- DH provides *only one* of the five desirable cryptographic services: secure key distribution.
- DH is commonly referred to as a *key agreement protocol*.
- DH is based on mathematics described as *calculating discrete logarithms in a finite field*.
- DH does not perform any kind of authentication of the endpoints.



When using Diffie-Hellman, the two participants use public keys and private keys to generate the same symmetric session key simultaneously in essence, providing secure key distribution. This can be accomplished across any communications channel, even when an attacker might be eavesdropping, because they only need to send public keys. Although the bad guy will see these public key values (remember that keys are nothing more than numbers), without knowing the private keys of the two participants, the bad guy cannot produce the same symmetric session key.

Here is how Diffie-Hellman works:

- 1.** BoBo and LuLu have no prior relationship and have never communicated previously. Today however, their paths cross, and they need to exchange confidential information securely over an unsecured communications channel.
- 2.** BoBo and LuLu agree on a public key value. Anyone monitoring their communications channel can see this number.
- 3.** BoBo calculates a mathematically related private key based on the agreed-upon public key value.
- 4.** At the same time, LuLu calculates a mathematically related and different private key based on the agreed-upon public key value.
- 5.** BoBo performs math by using his private key value and the agreed-upon public key value (Pub1). This produces BoBo's public key 2 value (BoBo Pub2).
- 6.** LuLu performs math by using her (different) private key value and the agreed-upon public key value (Pub1). This produces LuLu's public key 2 value (LuLu Pub2).
- 7.** BoBo sends BoBo's public key 2 value (BoBo Pub2) to LuLu. Anyone monitoring their communications channel can see this number.
- 8.** LuLu sends LuLu's public key 2 value (LuLu Pub2) to BoBo. Anyone monitoring their communications channel can see this number.
- 9.** BoBo performs math by using his private key value and LuLu's public key 2 value (LuLu Pub2). This produces BoBo's copy of the symmetric session key.
- 10.** LuLu performs math by using her (different) private key value and BoBo's public key 2 value (BoBo Pub2). This produces LuLu's copy of the symmetric session key, the same symmetric session key that BoBo just produced (see Figure 3-31).

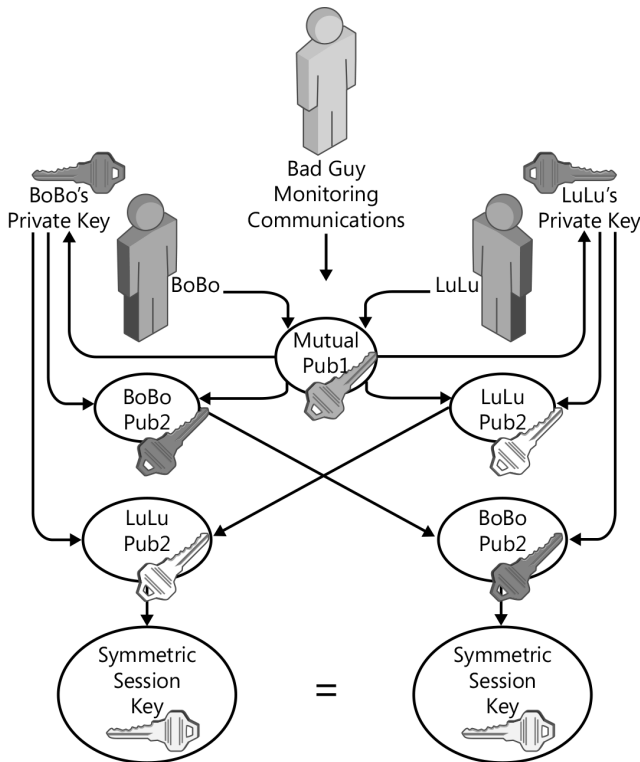


FIGURE 3-31 The Diffie-Hellman-Merkle algorithm

BoBo and LuLu now have copies of the same symmetric session key and can use these keys to encrypt and decrypt content to one another. The freshly and independently calculated symmetric session keys never commuted through the unsecured communications channel, so they remain secure.

An attacker could see three public keys commute through the unsecured communications channel—Pub1, BoBo Pub2, and LuLu Pub2—but without knowing either of the two private key values BoBo and LuLu used, she cannot calculate the same symmetric session key value (in a reasonably useful period).

The Diffie-Hellman algorithm can be used to calculate different lengths of symmetric session keys. The different length keys produced by DH are categorized by DH group ID numbers and are defined in IETF RFCs 2409, 3526, and 4492.

DH is most commonly used when digital certificates (PKI) are not present in the information environment. DH is a competing technology with other asymmetric, secure key distribution algorithms.

As stated before, Diffie-Hellman does not perform any kind of authentication of the endpoints, the two entities who want to establish symmetric keys. This vulnerability lends itself to identity spoofing (someone claiming to be someone he is not) and the man-in-the-middle

attack, when an imposter convinces two legitimate endpoints that she is the other endpoint. When identity spoofing and man-in-the-middle attacks are successful, the bad guy can affect the confidentiality, integrity, and availability of the valuable information assets of an organization.

This indicates that before you perform a Diffie-Hellman key exchange with another endpoint, you should strongly authenticate the other endpoint by using some other means of authentication.

RSA

While at the Massachusetts Institute of Technology (MIT), Ron Rivest, Adi Shamir, and Leonard Adleman, the founders of RSA, developed the RSA asymmetric key algorithm. The RSA company is now a division of EMC Corporation. The algorithm was introduced in 1978 and has become one of the most widely used asymmetric key algorithms. It is the preferred algorithm for PKI digital certificates used for signing and sealing content in mature and well-developed information systems.

- The RSA algorithm can be used to provide all five desirable cryptographic services.
- RSA is based on the difficulty of *factoring giant numbers into prime integers*.
- RSA keys typically range between 1 kilobit (1,024 bits) and 4 kilobits (4,096 bits) and are commonly used on PKI X.509 digital certificates.
- The RSA algorithm is commonly called a *trapdoor function* because the ease with which you can multiply many numbers to reach an answer, producing the public key, is like going up a ladder through a trapdoor into an attic compared with the difficulty of finding the closed trapdoor in the floor to come down the ladder from the attic (unless you know the secret).

The algorithm was patented in 1983 but was released into the public domain in September 2000. There have been cracks of up to a 786-bit key (they have been factored), but the use of these relatively short key lengths has been deprecated.

Elliptic Curve Cryptography (ECC)

Introduced in 1985 simultaneously by two mathematicians, Neal Koblitz and Victor Miller, this asymmetric key algorithm is based on pairs of numbers plotted on an elliptic curve and calculates discrete logarithm functions. ECC has the beneficial characteristic of using a relatively short key length and providing similar cryptographic strength to much longer keys used in other asymmetric key algorithms. According to the IETF in its RFC 4492 published in May 2006, a 163-bit key used in ECC has similar cryptographic strength to a 1,024-bit key used in the RSA algorithm. A 233-bit key used in ECC has similar strength to a 2,048-bit key used in RSA, and so on. The shorter keys used in ECC are much more efficient with CPU usage and therefore with power consumption. The ECC algorithm naturally finds a home in battery-operated or handheld devices, in which battery life can be extended without giving up cryptographic strength. This includes mobile phones, PDAs, and smart cards.

Because of this reduction in key length and the resulting reduction in computational effort required, ECC can be used to provide confidentiality for bulk content without the use of symmetric key cryptography for its performance benefits.

- ECC can provide all five desirable cryptographic services.
- ECC is based on pairs of numbers plotted on an elliptic curve.
- Keys typically range between 163 bits and 571 bits.
- ECC is desirable for its high speed, low processing power, low energy consumption, and low memory consumption.
- ECC is the preferred asymmetric algorithm for handheld and battery-operated devices.

NIST recommends that an ECC key should be at least twice the length of a symmetric key to provide equivalent strength. In FIPS-186-2, the US National Security Agency has approved the use of ECC-256 with SHA256 for protecting classified SECRET content and the use of ECC-384 with SHA384 for protecting classified TOP SECRET content.

ElGamal

ElGamal was published in 1984 by Taher ElGamal. ElGamal uses two schemes, one to provide encryption and another to provide a digital signature. It is used in the commercially available PGP asymmetric cryptosystem and in the free GNU Privacy Guard. The ElGamal Signature Scheme was used as the basis of the NIST Digital Signature Algorithm (DSA) used in the NIST-recommended DSS.

- ElGamal is an extension of the Diffie-Hellman algorithm and is based on mathematics described as calculating discrete logarithms in a finite field.
- ElGamal (the two schemes) can provide all five desirable cryptographic services.

ElGamal has a tendency to inflate the size of a message as it converts plaintext into ciphertext, often at a 2:1 ratio.

Digital Signature Standard (DSS)

In August 1991, NIST introduced the DSS based on the underlying DSA. It was adopted in 1993 as the Federal Information Processing Standard (FIPS) 186, which as of 2009 is on its third revision, FIPS 186-3. DSA is a variant of the ElGamal Signature Scheme. The DSA algorithm was patented in 1991 by two entities, Claus Schnorr (disputed) and David Kravitz of the NSA.

Recall that a digital signature relies on asymmetric key cryptography and a hashing algorithm. Because NIST recommends the use of its SHA family of hashing algorithms, it is easy to understand why SHA is a component of DSA and DSS. Remember that today SHA1 is not recommended, and the SHA2 family has become the NIST recommendation.

Several asymmetric key algorithms have been approved for use within DSA and DSS, including ElGamal, ECC, and RSA.

- DSA and DSS can be used to provide only digital signatures, only three of the five desirable cryptographic services.

- DSA can use different approved asymmetric algorithms but requires the NIST-approved SHA family of hashing algorithms.



EXAM TIP

The DSS can provide three desirable cryptographic services. They are strong authentication, strong nonrepudiation, and strong integrity validation.

LUC



LUC is a relatively fast asymmetric key algorithm. The name is from the mathematics on which the algorithm is based, called *Lucas sequences* (a family of mathematical functions that produce related integers, such as the Fibonacci series [1, 1, 2, 3, 5, 8, 13, 21 . . .]). LUC adds to this the calculation of discrete logarithms (such as those used in Diffie-Hellman). LUC can use relatively short keys (about half the size of an RSA key) and produce relatively strong ciphertext (compared to RSA). ECC is faster and similarly strong, so LUC gets little attention.

XTR

XTR gets its name from Efficient and Compact Subgroup Trace Representation (ECSTR). If pronounced, “ECSTR” sounds like “XTR.” XTR uses mathematics very similar to LUC, but through the addition of another specialized function, it can reduce the key length to about one-third the length of an RSA key while maintaining strength similar to RSA. ECC is faster and similarly strong so, like LUC, XTR gets little attention.

Knapsack

First introduced by Ralph Merkle and Martin Hellman, this algorithm is based on the much older knapsack problem. The problem poses a set of items, each with a specified weight and value, and a fixed-size knapsack to be filled with the best combination of the items based on the value versus the weight of the items. This problem is implemented mathematically to use a public key and a private key for contemporary cryptosystems. Unfortunately, several vulnerabilities have been identified in this family of algorithms, so it has fallen out of favor in recent years.



Quick check

1. Which asymmetric key cryptosystem is based on pairs of numbers on a plotted curve?
2. What services does the Digital Signature Standard provide?

Quick check answers

1. Elliptic Curve Cryptosystem
2. Authentication, nonrepudiation, and integrity validation

Cryptography in use

By now, you should have a solid view of how cryptography works, but where and how does cryptography fit into a contemporary information system? Remember that the primary objectives of a cryptosystem are to provide the following security services, ideally in a cost-justified manner, balancing the need for security (required protection for the valuable information assets) with the cost of implementation.

- Confidentiality
- Authentication
- Nonrepudiation
- Integrity
- Secure key distribution

This balance links to the risk assessment and information classification covered in Chapter 1 and the resulting policies that dictate what satisfactory protection is for each data element based on that classification. These policies define the strength of the security controls to be implemented to protect these assets while they are in storage (data at rest) and when they must be sent to some other recipient or location for use (data in transit).

A significant component of the cost of implementation is the price paid for slow performance of cryptographic services. Generally speaking, the strength of a cryptosystem is inversely proportional to its performance. In most cases, as strength gets better, performance gets worse, and you pay a higher price for the stronger security. Pay the minimum sufficient amount for the appropriate level of security as required by the data's classification and the company's policy.

You need to protect content while it is at rest and while it is in transit. Choose and implement the technologies that provide the needed service or services at the appropriate level of strength. This section addresses the various commonly used technologies to satisfy these issues in contemporary information systems.

They include:

- Link encryption
- End-to-end encryption
- Public-key infrastructure (PKI) with X.509 digital certificates
- Pretty Good Privacy (PGP)
- Secure communications channels for local area network–based (LAN-based) applications in the form of a virtual private network (VPN)
- Secure communications channels for web-based applications
- Steganography

Link encryption

Link encryption is a class of (typically) symmetric key encryption technologies used to protect data in transit. It is commonly implemented when the source and destination systems exist within two relatively trusted IT environments but these two locations must connect over a communications mechanism that cannot be properly secured otherwise (see Figure 3-32).

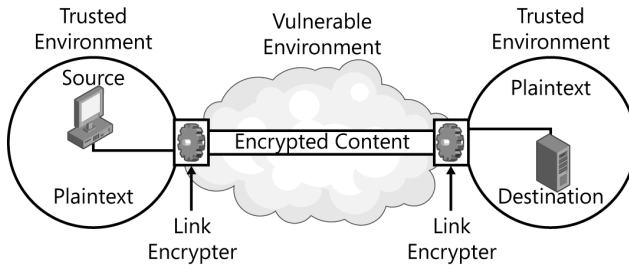


FIGURE 3-32 Link encryption

Link encryption devices are almost always configured with symmetric keys (which are faster for bulk encryption). You can think of link encryption as similar to a garden hose. What goes in one end is encrypted, making it unreadable, and it can only be decrypted (become readable) where it comes out at the other end. Nothing is readable in the middle. Because all content going into the link-encrypted channel becomes unreadable, there can be no intermediate infrastructure devices, or hops, that need to be able to read forwarding or routing information, and no error detection or correction can be performed, except at the endpoints of the encrypted channel or beyond the endpoints, on the trusted network.

If there are infrastructure systems that must read header or trailer information from the encrypted frame, those systems must decrypt the entire frame, read the required information, and then re-encrypt the frame. The infrastructure devices would need the symmetric key, and the content becomes exposed at the infrastructure device. This is usually not good, so if the connection between trusted networks does have to read frame data for forwarding, routing, or error detection, or for any other reason, link encryption is probably not the best solution and should be avoided. End-to-end encryption (described in the following section) should be considered for when infrastructure systems need to be able to read framing data in transit.

Examples of communication systems that might use link encryption include point-to-point T1 and T3 lines (described in more detail in Chapter 7, “Telecommunications and network security”) and telephone lines—for example, to secure sensitive fax transmissions. Another use for link encryption is to secure satellite communications channels. A link encryption device is placed inline between an Earth-based IT network and the uplink satellite dish. All communications are encrypted before transmission to the satellite. The satellite retransmits the encrypted content back down to Earth, where hundreds or thousands of antennas within the satellite’s footprint receive the signal. Only the link encryption device at the intended and authorized Earth-based destination IT network has the correct symmetric key to decrypt the

encrypted satellite retransmission. The plaintext content is then forwarded to the destination node on the intended, trusted network.

Notice that the information is in plaintext from the source to the uplink link encryption device and from the downlink link encryption device to the destination and is therefore potentially vulnerable. This exposure of plaintext data should be considered to ensure compliance with the policy of the organization.

End-to-end encryption

End-to-end encryption is a class of (typically) symmetric key encryption technologies that encrypts content at the source and decrypts it only at the destination, so that the protected data is never in plaintext while in transit. This helps ensure that only the authorized and intended recipient can access the protected content.

For this end-to-end encrypted content to make it over a standard routed network, forwarding information (a Layer 2 header), routing information (a Layer 3 header), and error detection information (used by routers and the recipient at Layer 2 but appended as a frame trailer) must remain in plaintext. Therefore, end-to-end encryption typically begins encrypting at the Layer 4 (Transport layer—also called the end-to-end layer) header of the standard Ethernet frame and stops at the end of the payload, just before the cyclic redundancy check (CRC) trailer used for error detection at each *hop* (router). This encryption structure is shown in Figure 3-33.

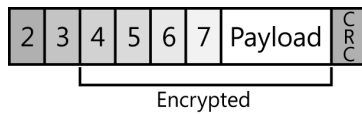


FIGURE 3-33 End-to-end encryption

End-to-end encryption is commonly used today in secure channel technologies such as IPsec, VPN, and SSL used over the Internet.



EXAM TIP

Understand the differences between, and uses for, link encryption and end-to-end encryption.

Public key infrastructure



The *public key infrastructure (PKI)* is a specific implementation of a hybrid cryptosystem. A PKI combines the functionality of symmetric, asymmetric, and hashing algorithms with the trust that can be provided by X.509 digital certificates in a hierarchical trust structure. The X.509 digital certificates and the PKI were developed by the ITU-T to provide strong security services within an X.500 directory service environment. A PKI can provide all five desirable cryptographic services in a strong manner. When strong security matters, PKI is the

recommended solution. However, it is an expensive solution, and it requires a large amount of administrative effort to establish and maintain. PKI is strong enough to have become the basis of assignment of formal legal liability in court.

Digital certificates from PKI systems are commonly used to provide user authentication of smart cards and system authentication on corporate networks and the Internet. They are used to provide SSL/TLS channels for many networking applications and protocols such as secure email, secure file transfers, and HTTPS to support secure e-commerce. They are used in dialup, wireless, and VPN authentication schemes such as the Microsoft Lightweight Extensible Authentication Protocol (LEAP) and Protected Extensible Authentication Protocol (PEAP).

The formation of a viable PKI begins with policies and procedures that define the structure and requirements of the PKI. Then technology (PKI-related hardware and software) must be implemented and specific roles and assignments of responsibility must be made. Users must be trained on the secure and approved use of the PKI and related technologies.

The PKI is implemented on computers or other specialized computing devices called certification authorities (CAs). These are the systems that create the digital certificates, binding a user identity to a public key. That public key is mathematically related to the private key accessible *only* to the aforementioned user, the user named on the digital certificate. The CA then digitally signs the digital certificate to provide strong proof that the CA created the certificate and that the certificate has not been altered or tampered with since it was created. The digital signature provides authentication so strong that it provides nonrepudiation and strong integrity validation at time of use.

The systems within a PKI trust the CA, and if the CA creates and issues a digital certificate that states that a user is BoBo, for example, all systems in the PKI trust that the user really is BoBo. As stated previously, in many cases, this declaration by the CA becomes a legally binding statement.

When the user submits the certificate as proof of identity to a system and the system validates the certificate, the system trusts the information written on the certificate and trusts that the public key embedded in the certificate belongs to the user named on the certificate. Now that user's public key can be used to accomplish secure and trusted cryptographic functions and communications, such as signing and sealing content, as needed in an IT environment.

The certification authority (CA)

The certification authority, and the hierarchy of CAs, is the structure of trust within a company. CAs are implemented to satisfy three roles within the PKI hierarchy:

- The root CA
- Policy CAs, subordinate CAs
- Issuing CAs, subordinate CAs

The *root* CA is the pinnacle of trust for an organization (or for the PKI). There is typically only one root CA within an organization. This system is so important to the PKI that to avoid



the possibility of compromise, in most implementations, the root CA never connects to any network, operating as a stand-alone system for its entire useful lifetime. The root CA is only rarely needed for service, so the system is usually powered off and is stored in a secured location. The root CA is used to generate subordinate CA certificates that authorize policy and issuing CA systems. These subordinate CA certificates must be sneaker-netted to the subordinate systems by using removable media, such as a floppy disk, writable optical media, or a USB thumb drive. This removable media must be verified to be free from malware before use. The root CA is certified by issuing itself a self-signed certificate.



A *subordinate CA* is any CA other than the root CA. Subordinate CAs can be policy CAs or issuing CAs. They are certified by a CA from a higher tier in the PKI hierarchy.



Policy CAs define a branch of the PKI hierarchy intended to comply with a business partner's PKI requirements. When the business partner—whether it is a supplier or a customer—is considered important enough and relies on PKI, companies will often commit the resources and add a branch to their internal PKI specifically to facilitate trusted business transactions with the partner. All CAs within the branch defined by the policy CA will be strictly managed following the policies and procedures declared on a certificate practices statement (CPS). The statement should match or exceed the PKI-related requirements of the intended partner. The CPS is usually published so that the intended partner can review the certificate practices promised by the company to ensure compliance with the partner's requirements and needs. The policy CA then is used to create one or more subordinate CA certificates for (typically) the issuing CAs within that branch governed by the stated policies. The policy CAs are kept online.



Issuing CAs are the systems that generate the X.509 digital certificates that are issued to the end entities. End entities include users, computers, and, more often these days, infrastructure systems such as routers, switches, and firewall appliances. These end entities are often called the subscriber or the subject of the certificate. This type of CA is accessed most often as users request and are issued certificates and when any certificate or certificates that were issued by the CA need to be revoked. The issuing CAs are kept online.

These CA systems should be hardened and dedicated-purpose systems, performing only the collection of CA functions. They must be protected from compromise. If attackers can compromise a CA, they would be able to generate digital certificates that certify a level of trust by the company that owns the PKI for themselves. They could claim any identity and have it certified by the company on the digital certificate. This places the company in a position of liability, and all users in the PKI and any trusting PKI would automatically trust the attackers' claimed but illegitimate identity, blindly facilitating their unauthorized and likely malicious activities.

It is generally considered that a PKI hierarchy should be at least two tiers deep, and most are three or four tiers deep. Multiple CAs are recommended at each tier and in each branch to provide redundancy, capacity, and perhaps geographic diversity to provide a local CA for faster and more reliable access.

An example of a relatively mature three-tier PKI hierarchy is shown in Figure 3-34.

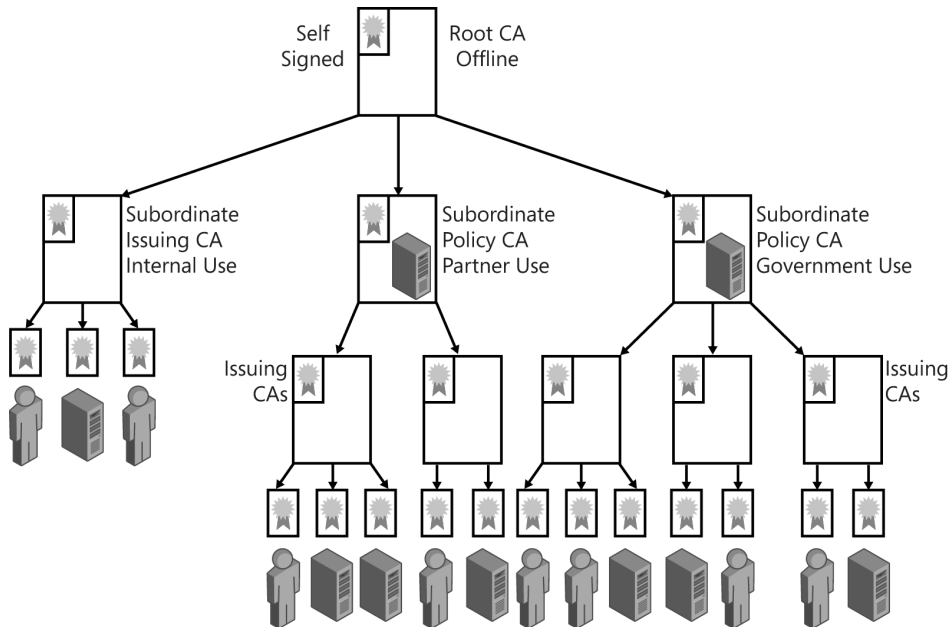


FIGURE 3-34 A PKI hierarchy

The registration authority (RA)

With all this formalized trust happening within a PKI, the issue of authenticating the user who is requesting a digital certificate surfaces as a critical issue. The PKI accommodates this with the use of a registration authority (RA). It is the job of the RA to authenticate end entities satisfactorily and attest to the CA that the required proof has been received and validated. The levels of required proof vary based on company policy and the class of certificate being requested. A higher value class of certificate is intended to protect higher value information assets, and the authentication requirements for those are more stringent. Certificate classes range from 1 to 5. For example, VeriSign uses the following descriptions for its certificates:

- Class 1 for individuals, intended for email
- Class 2 for organizations, for which proof of identity is required
- Class 3 for servers and software signing, for which independent verification and checking of identity and authority is done by the issuing certificate authority
- Class 4 for online business transactions between companies
- Class 5 for private organizations or governmental security

Companies might choose to use these class descriptions, use their own classes, or reference no classes on their certificates. The following list shows some examples of different levels of user authentication that might be required within a PKI system:

- A basic level is successful user authentication in directory services so that the authentication server trusts the identity claim of the user. The authentication server is accepted as the RA. Another example is a supervisor's declaration of the user's identity by using an official communication mechanism. In this case, the supervisor acts as the RA.
- In-person authentication requires the user to complete and sign an application form and present to an authorized RA (a person) one federal government official photo ID, such as a Department of Defense (DoD) ID or a passport, or two nonfederal issued IDs such as a driver's license and birth certificate.
- Add to the in-person authentication an interview with the (human) RA in which cognitive questions are asked from an on-file profile of the user to verify identity further.
- Add to that a biometric authentication mechanism.
- Add to that a DNA verification.

Trusting a certification authority or PKI



Subjects participating within a PKI establish trust for the PKI by importing the digital certificate from the root CA into their *Trusted Root Certification Authorities Store*. Importing a root CA certificate from a different organization's PKI provides cross-certification to establish trust for the other organization. Importing a digital certificate from a nonroot CA from a different organization's PKI establishes subordinated trust for that CA and its subordinates from the other organization.

By acquiring and storing the certificate from a CA, a system now has access to the public key for the CA and can validate the CA's digital signature attached to every certificate the CA issues. This strongly proves to the system that a certificate was issued by a trusted CA and that the certificate has not been tampered with since it was created and issued by the trusted CA. The system can now trust the information presented on the digital certificate and can trust that the public key embedded on the digital certificate belongs to the subject named on the certificate. This facilitates the signing and sealing processes required by the PKI-enabled information system.

Typically, the provider of the operating system (such as Microsoft) automatically adds the collection of globally recognized trustworthy public CAs to the Trusted Root Certification Authorities Store to provide PKI functionality right out of the box for an operating system. If you decide you don't want to trust a CA or PKI, simply remove that CA's certificate from the Trusted Root Certification Authorities Store.

NOTE THE TRUSTED ROOT CERTIFICATION AUTHORITIES STORE

An easy way to access the Trusted Root Certification Authorities Store on a computer with a Microsoft-based operating system is to open Internet Explorer. On the menu bar, select Tools | Internet Options. Click the Content tab and then click the Certificates button. Click the Trusted Root Certification Authorities tab.

The X.509 digital certificate



The *X.509 digital certificate*, currently in version 3, is an *International Telecommunication Union Telecommunication Standardization Sector (ITU-T)* standard format for certificates and certificate revocation and validation. It was introduced in July 1988 as an additional component to provide strong security services for the X.500 Directory Services standard. In 1999, the IETF adopted the X.509 standard in RFC 2459 and has updated that with the current RFC 5280 (issued in 2008), often called PKIX (PKI X.509).

The X.509 standard defines the following fields for the digital certificate:

- Certificate
- Version
- Serial Number
- Validity
 - Not Before
 - Not After
- Algorithm ID
- Issuer
- Subject
- Subject Public Key Info
 - Public Key Algorithm
 - Subject Public Key
- Issuer Unique Identifier (optional)
- Subject Unique Identifier (optional)
- Extensions (optional)
- Certificate Signature Algorithm
- Certificate Signature

NOTE WHAT ALGORITHMS TO USE

Many of the PKI cryptographic functions today rely on the RSA asymmetric algorithm and SHA1 and the SHA2 family of hashing algorithms for signing and sealing. The symmetric algorithm is identified in the application itself and not in the digital certificate; however, AES is the prevalent choice today.

The following describes a typical commercial scenario:

1. The subject presents a claim of identity and a certificate request to the RA.
2. The RA validates the subject's identity by following the authentication standards in place for the organization and, based on the type of certificate being requested, typically defined in the policies of the organization.
3. After the RA has successfully completed the verification process, the RA requests the issuance of the certificate by the CA on behalf of the subject.
4. If the certificate has been authorized by management for the subject, the CA requests the public key from the subject.
5. On the subject's local computer, this triggers a call to the cryptographic service provider (CSP) that generates keys, matching the key requirements specified by the CA for the requested certificate. The CSP generates a public key/private key pair and securely stores the private key in the local computer's key store.
6. The subject's system then sends the public key to the CA.
7. The CA binds the details of the subject, the details of the certificate, and the public key into the digital certificate.
8. The CA digitally signs the digital certificate by generating a hash value of the certificate, including the subject's public key, and encrypting the hash value by using the CA's private key.
9. The certificate is then issued to the subject and is generally forwarded to the certificate repository for the PKI-enabled application where the digital certificate is to be used.

Figure 3-35 details this process.

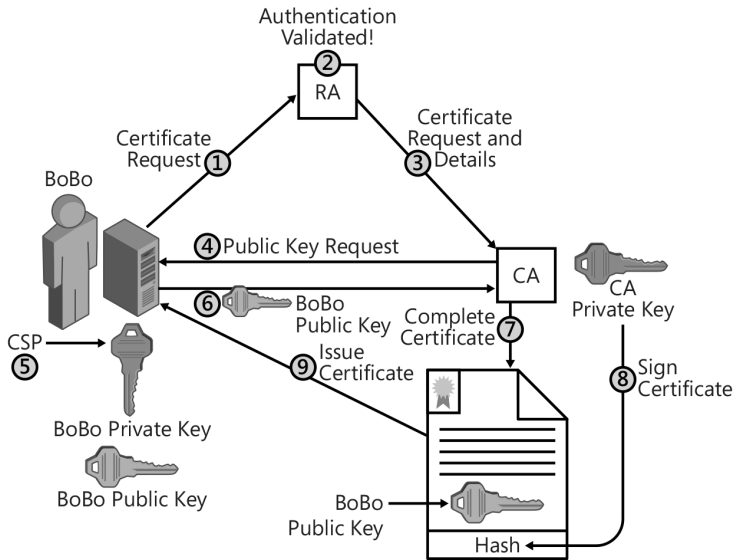


FIGURE 3-35 Acquiring an X.509 digital certificate

In government use, it is common for a CSP on the CA to generate the private key and public key pair. The private key is then encrypted on the *common access card* (CAC) smart card, and the public key is bound to the digital certificate. The digital certificate might also be copied to the CAC card, but it isn't typically necessary to encrypt the digital certificate with the public key.

Digital certificates usually specify an intended purpose. The following is a list of the commonly used intended purposes:

- Client authentication
- Server authentication
- Smart-card logon
- Secure email
- Code signing
- IPsec/IKE intermediate
- KDC authentication
- Certificate request agent
- Private key archival
- Key recovery agent
- Online Certificate Status Protocol (OCSP) signing
- Microsoft Trust List signing (Microsoft systems)

An example of how an X.509 v3 digital certificate is represented on a computer is shown in Figures 3-36, 3-37, and 3-38.

If there is a certificate practices statement from a higher-level policy CA, the Issuer Statement button, which is unavailable in Figure 3-36, will be active and link to that policy statement.

The Copy To File button shown in Figure 3-37 allows the export of the digital certificate for manual distribution on removable media or through email or some other networking file copy function.

If the certificate shown in Figure 3-38 had been issued by a subordinate CA, the Certification Path tab would show the chain from the issuing CA to the root CA of the PKI.



FIGURE 3-36 A sample X.509 digital certificate showing the General tab



FIGURE 3-37 A sample X.509 digital certificate showing the Details tab



FIGURE 3-38 A sample X.509 digital certificate showing the Certification Path tab

When a certificate is presented to initiate a secure cryptographic function, the server should validate the certificate before trusting the certificate. This is a process written into the PKI-enabled application. The higher the value of the information assets that the certificate is intended to protect, the more strictly the certificate should be verified and the more likely the program should be to reject the requested protected function if the certificate fails to validate completely and successfully. Certificate validation checks should include the following:

- Verifying that the certificate was issued by a CA hierarchy that is trusted
- Verifying the digital signature of the certificate to prove the source CA of the certificate and the integrity of the certificate
- Verifying that the details of the certificate match the nature of the request, such as that the name of the subject on the certificate matches the name of the requester, or the name of the web server, and that the requested function matches the certificate's intended purpose
- Verifying that the certificate hasn't expired, as shown in the Valid From and Valid To dates
- Verifying that the algorithms and key lengths satisfy the company's policy and security requirements for the type of transaction requested
- Verifying that the certificate revocation list (CRL) is accessible and that the certificate has not been revoked (discussed shortly)

Sometimes, to prevent a program from appearing to fail, a developer might allow the user to bypass a failed certificate validation, allowing the nonsecure process to execute and exposing the system and data to potential compromise. This defeats the whole purpose of implementing a PKI. Strong security should dictate that all certificates receive full validation checks, and any certificate validation failure fails the related process.

The certificate repository

When application developers enable PKI in their applications, the application server typically becomes the certificate repository for that application. Client-side applications natively connect to the application server for routine operations, making that server the most efficient distribution point for storing and accessing digital certificates for the participants using the application.

For example, if the certificates are used for user authentication within directory services, the directory service server becomes the certificate repository. If the certificates are used for signing and sealing email messages, the email server is designed as the certificate repository.

Certificate revocation

To limit the exposure and liability of an organization, it often becomes necessary to revoke certificates that have been issued, and the organization no longer wants those certificates to be honored if presented for use. An issued certificate must be revoked at the CA that issued the certificate. When an administrator revokes a certificate, the serial number for the certificate



is added to a *certificate revocation list (CRL)*. This CRL must be published so that systems that are presented with a certificate can verify whether that certificate has been revoked. The issued certificates include where the CRL is published. These locations are called *CRL distribution points (CDPs)*. A CDP location is often:

- A publicly accessible website.
- A location within directory services, accessible by a Lightweight Directory Access Protocol (LDAP) query.
- On a network share, accessible over the corporate network.
- On an FTP server for download, often accessed by partners participating with the organization's PKI.

According to RFC 5280, there are currently 10 reasons to revoke a certificate. (Although the list contains 11 items, 7 isn't used.) They are:

0. Unspecified
1. Key Compromise
2. CA Compromise
3. Affiliation Changed
4. Superseded
5. Cessation Of Operation
6. Certificate Hold
7. (not used)
8. Remove From CRL
9. Privilege Withdrawn
10. Compromise

The only one of these that is reversible is the Certificate Hold, which is typically used when a subject will not be participating in the PKI for some period of time, such as when going on sabbatical or an extended leave of absence.

Another technique to publish and access the CRL is by using the newer OCSP. This provides greater accessibility and faster response to revocation status queries.



EXAM TIP

The use of X.509 digital certificates can provide all five desirable cryptographic services in a strong manner.

Pretty Good Privacy (PGP)




Pretty Good Privacy (PGP), currently in its tenth version, is a commercially available hybrid cryptosystem that includes the use of asymmetric algorithms, symmetric algorithms, and hashing algorithms. PGP is a competing technology with PKI and provides most of the same

services and functions. It was created by Phil Zimmermann in 1991 and was published as an open standard called *OpenPGP* in the IETF RFC 2440 in 1998. The RFC was updated in 2007 in RFC 5581. PGP does not use X.509 digital certificates natively, but it has been updated to participate with and, to some level, integrate with the PKI system.

PGP provides services that include:

- Signing and sealing email messages and attachments (protection for data in transit).
- Whole-disk encryption (protection for data at rest).
- Scalable encrypted volumes (encrypted store to self) (protection for data at rest).
- PGP NetShare (encrypted store to authorized others) (protection for data at rest).
- Protection of instant messaging (IM) (protection for data in transit).
- Secure deletion (overwriting remnants).



The most significant difference between PKI and PGP is that PKI uses a hierarchical trust model, whereas PGP uses a less structured, *web of trust* model. This model has several vulnerabilities, including the need to verify authentication codes manually on the certificate form (not X.509) used by PGP to establish trust for the certificate. Another weakness is the issue of associative trust, by which user BoBo trusts user LuLu, and user LuLu trusts user Willis. If the trust is not managed strictly in PGP, user BoBo will have a level of trust for Willis without ever knowing Willis, trying to trust Willis, or even being aware of his trust for Willis.



The public key of the other PGP user is stored on a *key ring* versus a key store as in PKI.

NOTE A COLORFUL PAST

Zimmermann named PGP after a fictional business named Ralph's Pretty Good Grocery, a store in a radio program. The original algorithm was called BassOmatic, from a Saturday Night Live skit. BassOmatic is a 128+ bit symmetric key algorithm. The original version was distributed free and was provided by Zimmermann to hide antinuclear activist content on Bulletin Board Systems (BBS) and the Usenet.

In 1993, due to violations of US export regulations, Phil Zimmermann became the formal target of a criminal investigation for "munitions export without a license." Anything larger than 40 bits was considered to be weaponry and was restricted for export. In an act of defiance, Zimmermann published the entire source code in a book and continued to distribute the code because books are protected under the First Amendment and are therefore exportable. The code was designed to be scanned and then run through optical character recognition (OCR) software, resulting in the easy distribution of the source code. The federal investigation was eventually dropped after several years without any charges filed against anyone, ever.

PGP 2 bumped into patent violations with respect to the CAST, DSA, and ElGamal algorithms. PGP v3 and v4 bumped into patent violations with the RSA algorithm. These patent violation disputes were eventually resolved.

After going through several owners, PGP was purchased in 2010 by Symantec Corporation and is part of their Enterprise Security Group.

Secure channels for LAN-based applications

It is often necessary to protect content during transit between two endpoints over a corporate network or between different geographic network locations. In this scenario, the endpoints are often somewhat trusted, but the connection between these previously known and somewhat trusted endpoints is potentially hazardous and untrusted, such as a connection over the Internet between headquarters and a branch office. The technology commonly deployed to solve this problem is called the *virtual private network (VPN)*.

The VPN works by isolating the network frames from the surrounding networking environment, often referred to as *encapsulation*. VPN protocols are also often referred to as *tunneling protocols*. In most cases, the protection provides encryption and some form of authentication and integrity validation.

Following are several commonly used VPN technologies:

- Secure Shell (SSH)
- Point-to-Point Tunneling Protocol (PPTP)
- Internet Protocol Security (IPsec)
- Layer Two Tunneling Protocol (L2TP)
- Secure Socket Tunneling Protocol (SSTP)

Secure shell (SSH)

In the earliest days of networking, to avoid needing to be physically close to computers to perform maintenance and configuration changes, administrators used a protocol called *Telnet* to perform remote administration. However, Telnet performed its authentication in cleartext, and all information commuted on the wire in cleartext. In 1995, SSH was introduced as an encrypted channel to perform this authentication and remote administration securely.

Telnet uses TCP port 23, and SSH uses TCP port 22. SSH is an Open Systems Interconnection (OSI) Model Layer 7 protocol. Servers must run the SSH daemon (server service), and the administrative workstation must run the SSH client.

SSH uses asymmetric keys to authenticate users, and its implementation does not verify ownership of key pairs or identities bound to key pairs. It originally used IDEA for bulk encryption of the data and a 32-bit CRC for integrity verification. The most recent version, SSH-2, uses DSA, RSA, ECDSA, or X.509 digital certificates for authentication. It can use Diffie-Hellman for secure key exchange and Message Authentication Code (MAC) for authentication and integrity validation. Many implementations support the use of passwords (symmetric keys) for endpoint or user authentication.

Although not originally designed or intended to be used as a VPN, its inherent functionality allows for this. It is not considered to be the strongest encryption, but SSH has nonetheless become popular for use in securing FTP traffic in the *SSH File Transfer Protocol (SFTP)*.

Point-to-Point Tunneling Protocol (PPTP)

Point-to-Point Tunneling Protocol (PPTP) was designed to secure Point-to-Point Protocol (PPP) connections. Published in IETF RFC 2637 in 1999, PPTP uses TCP port 1723. The PPTP protocol doesn't specify any encryption technology but refers to the use of generic routing encryption (GRE). Microsoft implemented PPTP by using Microsoft Point-to-Point Encapsulation (MPPE), which uses RC4. Although it is widely used on Microsoft systems, versions are available for Linux and Macintosh operating systems.

With the relatively recent addition of the Extensible Authentication Protocol (EAP), Microsoft has included support for its Protected EAP (PEAP) for secure authentication. PPTP is considered relatively weak, and several vulnerabilities have been identified.

Internet Protocol Security (IPsec)

It is clear that TCP/IP has become the standard for Transport layer and Network layer protocols, and although the rest of the world has pretty much migrated on to IPv6, the United States is still largely operating on IPv4. IPsec was created to provide security capabilities for IPv4 network traffic, regardless of its higher layer source. IPsec has become the de facto VPN because of its strength and tremendous versatility. It is defined in the IETF RFC numbers 2401, 2402, 2406, 2408, and 2409 and uses UDP port 500.

NOTE SECURITY BUILT INTO IPV6

Internet Protocol version 6 (IPv6) has security features built into it and therefore does not require a different protocol, as IPv4 needs IPsec, for security.

In its default state, IPsec uses symmetric key cryptography and therefore can provide strong encryption and weak authentication and integrity validation. IPsec can be strengthened to provide the cryptographic services strongly through the addition of digital certificate-based authentication.

IPsec can provide host-to-host, host-to-subnet, or subnet-to-subnet VPN connectivity. These implementations are shown in Figure 3-39.

The use of different IP subnets causes the routing system on the LAN or LANs to forward the plaintext packets through the IPsec VPN gateway systems, where they are encrypted in the host-to-subnet and subnet-to-subnet designs.

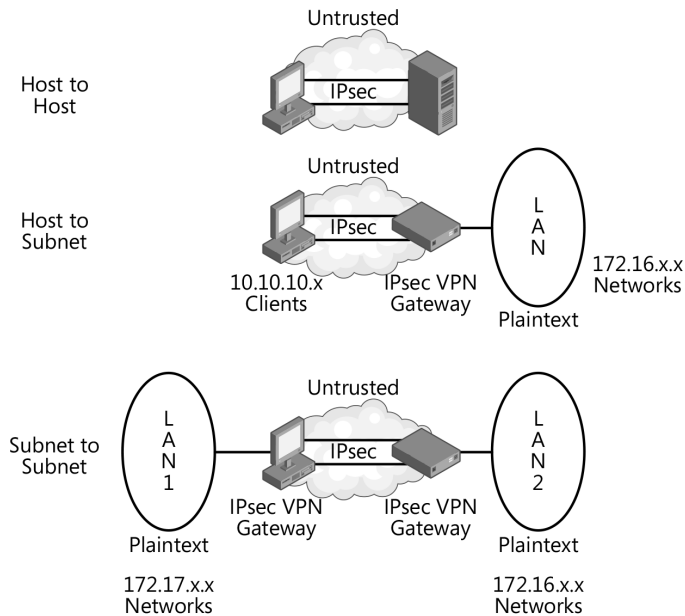


FIGURE 3-39 IPsec in host-to-host, host-to-subnet, and subnet-to-subnet implementations

INTERNET KEY EXCHANGE (IKE)

The IPsec VPN is highly customizable. Different types and levels of protection can be configured differently based on many issues, such as:

- Authentication type: Kerberos protocol, digital certificates, or a pre-shared key (such as a password)
- Direction: inbound/outbound/mirrored
- Protocol
- Port numbers
- IP addresses
- Data authentication and integrity verification (AH Mode)
- Data encryption (ESP mode; the default behavior is to provide AH + ESP for data)
- Encryption algorithm
- Key distribution technique
- Encryption key rotation frequency (typically every 100,000 KB or every 60 minutes, whichever comes first)
- Hashing algorithm
- Transport mode vs. Tunnel mode
- Perfect Forward Secrecy—On/Off

The list of options goes on and on.



These details are figured out and documented by a protocol called *Internet Key Exchange (IKE)*, currently in version 2. In the first phase of IKE, a protocol called *Internet Security Association and Key Management Protocol (ISAKMP)*, pronounced *isa-camp* establishes a secure channel, and then it establishes the framework for the negotiation of the details of the IPsec VPN. However, ISAKMP itself does not negotiate the details. IKE negotiates all the numerous details of the IPsec VPN and records the agreed-upon details in two lists called *security associations (SA)*. There is one SA with details for the inbound connection and one SA with details for the outbound connection. These two SAs are stored on each endpoint in a database. One endpoint's inbound SA is the other endpoint's outbound SA and vice versa. There might be as many as four SAs per IPsec VPN, one SA pair for the AH protocol and one SA pair for the ESP protocol. The AH and ESP protocols are covered shortly within this section. To improve performance and keep track of potentially hundreds of active VPN connections, these SAs are indexed, and the index is assigned a numeric value called the Security Parameter Index (SPI).



In the second phase of IKE, the *Oakley Key Determination Protocol* allows the two endpoints to agree on how to exchange symmetric keys securely for encryption and decryption.

After all the details are understood, it is almost time to start sending data packets through IPsec. But first, IKE must generate and distribute the symmetric encryption keys to be used for data authentication and encryption. This typically is accomplished by using Diffie-Hellman.



However, IPsec supports a protocol called the *Secure Key Exchange Mechanism (SKEME)*, which allows various key distribution techniques to be implemented such as the use of static and manually input symmetric keys or the use of public key/private key secure key distribution. As soon as the SAs are complete and the symmetric keys are distributed, IKE is done, and data packets can finally flow, protected by the security mechanisms just negotiated.

AUTHENTICATION HEADER (AH)



Authentication header (AH) is a security component of IPsec that performs symmetric key authentication and integrity validation but not encryption (confidentiality). The negotiated and agreed-upon details of AH are documented within a pair of SAs at each endpoint. AH runs a hashed message authentication code (HMAC) algorithm from the beginning of the Layer 3 header to the end of the payload to produce an integrity check value (ICV) for the IP packet, as shown in Figure 3-40.

AH Transport Mode Packet Structure



FIGURE 3-40 Authentication header (AH) in IPsec

The recipient validates this ICV to provide authentication and integrity validation.

Notice that the ICV is calculated over the Layer 3 header. This means that if the Layer 3 header data is changed, the ICV validation will fail, and the packet will be discarded by the

recipient. This is why IPsec and Network Address Translation (NAT) have potential conflicts that must be avoided. NAT is covered more in Chapter 7.

AH is defined as Protocol ID 51.

ENCAPSULATING SECURITY PAYLOAD (ESP)



Encapsulating Security Payload (ESP) provides symmetric key authentication, integrity validation, and encryption (confidentiality). The negotiated and agreed-upon details of ESP are documented within a pair of SAs at each endpoint. The encryption begins at the end of the Layer 3 header and before the Layer 4 header, and then a Message Authentication Code (MAC) value is calculated across a portion of the Layer 3 header and the entire encrypted payload of the packet, as shown in Figure 3-41.

ESP Transport Mode Packet Structure

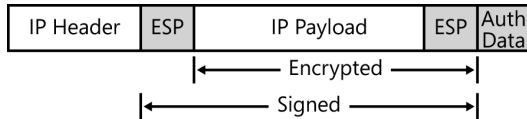


FIGURE 3-41 Encapsulating Security Payload (ESP) in IPsec

The recipient validates the ESP MAC value to provide authentication and integrity validation. ESP also provides a sequence number for anti-replay protection. Because ESP does not encrypt Layer 3 or include the source and destination IP addresses in its MAC calculation, ESP alone does not have any trouble passing through a NAT server. This is referred to as IPsec NAT-T (NAT Traversal).

ESP is defined as Protocol ID 50.

TRANSPORT MODE



The *Transport mode* of IPsec implements encryption at the beginning of the Layer 4 header, the Transport layer. This encryption continues until the end of the payload and before the 32-bit CRC trailer, as shown in Figure 3-42.

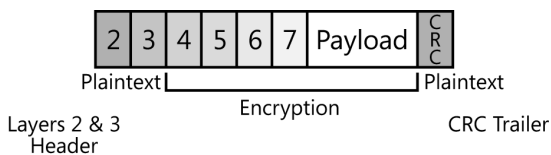


FIGURE 3-42 The Transport mode of IPsec

This allows the forwarding information (destination MAC address information) in the Layer 2 header, the routing information (destination IP address), and the 32-bit CRC trailer to remain in plaintext so that the network infrastructure systems can successfully transmit the packet to its destination.

The plaintext presentation of actual source and destination IP addresses reveals network architecture and might be considered too much information to provide to the unauthorized (attackers). If the security requirements of the organization dictate that this information must also be protected, Transport mode should not be used, but IPsec has an alternative that will satisfy these concerns.

TUNNEL MODE



To protect the actual source and destination IP addresses with encryption, run IPsec in *Tunnel mode*. This encrypts the entire packet from the beginning of the Layer 3 header to the end of the payload. Then it adds a new, plaintext Layer 3 header that contains the destination IP address of the IPsec VPN gateway system on the remote side of the IPsec VPN. Finally, at Layer 2, it adds a plaintext Layer 2 header and the CRC trailer to complete the frame. This leaves Layer 2 information, Layer 3 information, and the CRC trailer in plaintext, just enough information to pass the frame through a routed network. This is shown in Figure 3-43.

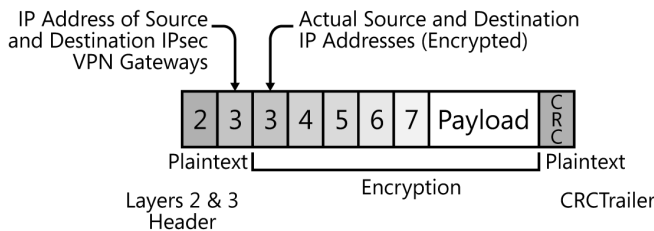


FIGURE 3-43 Tunnel Mode of IPsec

Tunnel mode protects the true source and destination IP addresses, hiding all indications of the internal corporate network architecture and indications of interesting targets (end-points) for any potential attackers who might just be watching the encrypted packets commute on the public networks.

Layer Two Tunneling Protocol (L2TP)



Published in 1999 as the IETF RFC 2661, *Layer Two Tunneling Protocol (L2TP)* was developed jointly by Cisco and Microsoft, merging Layer 2 Forwarding (L2F) Protocol and PPTP. L2TP was designed to tunnel many types of protocols from many applications using Point-to-Point Protocol (PPP) connections over IP (or other) networks to connect to an L2TP network server (gateway). Interestingly, L2TP does not provide any encryption. If confidentiality services are required, some additional encryption technology must be implemented. In common practice, IPsec is used to provide this encryption service. This is referred to as an *L2TP/IPsec tunnel*. When L2TP is used with IPsec, X.509 digital certificates for server authentication can be used. By using these certificates, L2TP/IPsec provides strong authentication of the endpoints in addition to strong integrity verification. The Encapsulating Security Payload of IPsec provides encryption for confidentiality. This implementation is shown in Figure 3-44.

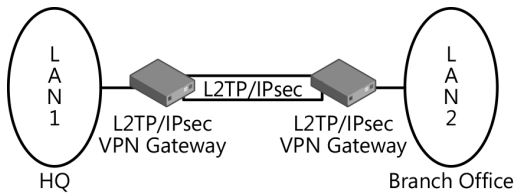


FIGURE 3-44 L2TP over IPsec

Although L2TP can work on an IP network, it can also be used on wide area network (WAN) Layer 2 networks such as X.25, frame relay, and Asynchronous Transfer Mode (ATM). L2TP uses UDP port 1701, but when it is used with IPsec implemented at the same gateway node, IPsec encrypts the L2TP frame and exposes only IPsec's UDP port 500.



EXAM TIP

The initiating protocols, authentication mechanisms, key exchange mechanisms, security protocols, and modes of IPsec should be understood for the exam.

Secure Socket Tunneling Protocol (SSTP)



Secure Socket Tunneling Protocol (SSTP) was introduced as an encrypting VPN technology in approximately 2008. SSTP is a client to the VPN server (gateway) encrypted channel designed to transport PPP or L2TP protocols securely over IP networks. SSTP requires a digital certificate on the server side. Client authentication is optional but can include no authentication, CHAP, and certificate-based authentication mechanisms when the EAP is used. SSTP establishes an SSL tunnel (described in the next section) and ports all encrypted traffic over TCP port 443, the same port used by HTTPS. This facilitates its use through corporate firewalls, in which HTTPS must already (typically) be allowed.

SSTP does not support subnet-to-subnet VPN connectivity.

Secure channels for web-based applications

There certainly is a need to protect information in transit over the Internet between client computers and web-based servers. In this arena, and in contrast to VPN technologies, there is a greater expectation that the endpoints of a web-based session are less trusted and are more likely to have no prior relationship. The security goals remain the same to provide confidentiality and some form of authentication and integrity validation. If authentication can be implemented strongly, you can get nonrepudiation, and because you will typically need to send bulk data, you will probably need to distribute symmetric session keys securely.

The following technologies are or have been commonly used to provide secure communications for web-based traffic:

- Secure Sockets Layer (SSL) and Transport Layer Security (TLS)
- Hypertext Transfer Protocol over Secure Sockets Layer/Transport Layer Security (HTTPS)
- Secure Hypertext Transfer Protocol (S-HTTP)
- Secure File Transfer Protocol (SFTP) and FTP over SSL (FTPS)
- Secure Electronic Transaction (SET)
- Secure Multipurpose Internet Message Extensions (S/MIME)

Secure Sockets Layer (SSL) and Transport Layer Security (TLS)

Secure Sockets Layer (SSL) is the predecessor to Transport Layer Security (TLS). SSL/TLS has become the de facto standard for protecting web-based content in transit. SSL was developed by Netscape and is defined in the IETF RFC 5246 (the latest update). Due to security flaws, the first stable version of SSL was version 3, published in 1996. TLS version 1 was released in 1999, and although it is very similar, it is an upgrade to SSL version 3, improving security and fixing flaws. TLS is currently in version 1.2, with its most recent update released in March 2011, IETF RFC 6126. SSL version 3.1 is approximately equivalent to TLS version 1.2. This release upgrades TLS to the use of AES with counter mode and SHA-256, and disallows the negotiation down to the weaker SSL version 1 and version 2. Almost all current browsers support TLS version 1, and Internet Explorer 8 and later supports TLS version 1.2.

SSL/TLS typically requires an X.509 digital certificate for server authentication (now often called an SSL certificate). This provides strong authentication of the server to the client, so the client can trust that it has connected to the correct remote system. (Figure 3-45, in the next section, shows the use of SSL/TLS to protect HTTP traffic [HTTPS]).

Hypertext Transfer Protocol over SSL/TLS (HTTPS)

Hypertext Transfer Protocol over Secure Sockets Layer/Transport Layer Security (HTTPS) might also be called Hypertext Transfer Protocol Secure. HTTPS was designed to provide strong authentication, nonrepudiation, confidentiality, and integrity validation services for HTTP-based traffic. HTTPS was invented by Netscape in 1994 and was accepted by Microsoft, being implemented in the two most popular browsers in their time. Although not a perfect cryptographic protocol, HTTPS has risen to the top of the list to provide strong cryptography for HTTP traffic and web-based applications. HTTPS is defined in IETF RFC 2660 and typically uses port 443 for its HTTPS traffic.

HTTPS requires a digital certificate for server authentication (often referred to as an SSL certificate) on the HTTPS web server. The establishment of the SSL/TLS tunnel is described in the following steps (see also Figure 3-45):

- 1.** The client initiates a session with the web server. This is often done using the HTTP-Get command sent to the server's port 80.
- 2.** The server is configured to run HTTPS and notifies the client of the protocol change. Cryptographic algorithms are negotiated, and the server sends its X.509 server certificate to the client.
- 3.** The client validates the server's X.509 digital certificate (confirming a trusted CA, that the digital signature validates, that the details on the certificate are accurate and match the session, that the certificate is not expired or revoked, and so on). After the certificate is validated, the client trusts the certificate and that the embedded public key belongs to the server whose name appears on the certificate (strong server authentication and nonrepudiation).
- 4.** The client system activates the correct cryptographic service provider (CSP) on the client system to generate a pair of symmetric session keys that satisfy the cryptographic algorithm or algorithms negotiated earlier.
- 5.** The client uses the server's public key from the digital certificate to encrypt one of the newly created symmetric session keys.
- 6.** The client sends the encrypted symmetric session key to the server (secure key distribution).
- 7.** The server uses its private key to decrypt the symmetric session key from the client.
- 8.** The client and server now switch to the HTTPS protocol. The client switches to destination port 443 for its outbound traffic to the server. The client and server now encrypt everything outbound in the session and decrypt everything inbound in the session (strong confidentiality). Most implementations include the use of some form of message authentication code (MAC) (integrity validation).

Depending on the nature of the services provided on the web server, client authentication might be required by the server. This authentication can be negotiated and performed within the secure SSL/TLS channel. In a Microsoft implementation, this is called Protected Extensible Authentication Protocol (PEAP). In a Cisco implementation, this is called Lightweight Extensible Authentication Protocol (LEAP).

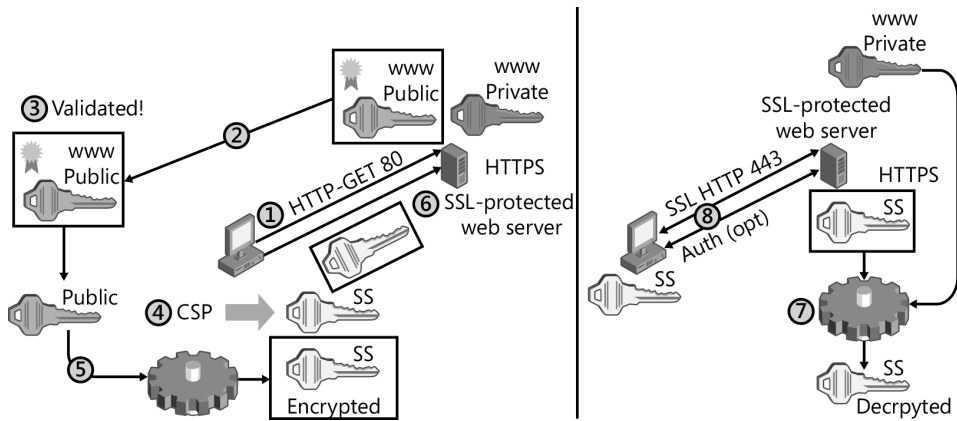


FIGURE 3-45 Hypertext Transfer Protocol over Secure Sockets Layer/Transport Layer Security (HTTPS)

On a publicly accessible web server running HTTPS, the digital certificate is typically acquired from an external public certification authority (CA) such as VeriSign, Entrust, Equifax, GeoTrust, or Thawte. To support internal users, for example for user authentication within a directory services environment, the certificate can be generated by an internal, private, but trusted CA.

Secure Hypertext Transfer Protocol (S-HTTP)

Secure Hypertext Transfer Protocol (S-HTTP) and HTTP over SSL (HTTPS) are competing technologies used to secure web-based traffic. They were both introduced in the mid-1990s and can use similar encryption, authentication, and hashing algorithms. Both support the use of digital certificates. However, S-HTTP performs a message-based encryption, whereas HTTPS encrypts the entire channel, providing session-based encryption.

S-HTTP is defined in IETF RFC 2660 and can use the same port 80 for its traffic. S-HTTP hasn't been accepted nearly as well as HTTPS and has largely fallen by the wayside. S-HTTP was designed to be very flexible, allowing many options for application developers to design S-HTTP into their applications. This has been likened to giving the developers enough rope to hang themselves. Due to the lack of strict specification, application developers, who very often are not cryptographers, can implement the cryptographic protocols improperly, allowing the potential for massive vulnerabilities.

Secure File Transfer Protocol (SFTP) and FTP over SSL (FTPS)

The File Transfer Protocol was one of the early networking protocols designed to transfer bulk content by using guaranteed delivery TCP. FTP includes user authentication but performs that authentication in plaintext. This is a bad thing. Plaintext authentication is never preferred because many users use their directory services logon account user names and passwords for the FTP server so that they don't have to remember so many credentials. This exposes their corporate logon credentials. SFTP and FTPS were invented to help resolve this problem.

SFTP uses an SSH connection and then runs the FTP protocol through the secure shell. The latest version of SSH is version 2. The FTP client software must support the SFTP protocol. SFTP typically runs on SSH port 22. An early and fallback technology uses the UNIX/Linux secure copy (*scp*) utility.

FTPS establishes an SSL secure channel and then runs the FTP session through SSL. Both FTPS and SFTP seem to be getting business, and current implementations of both are common. IANA assigned ports 989 (data) and 990 (control) for FTPS, but vendors often use custom port numbers. Remember that SSL requires the use of a digital certificate from a trusted CA.

Secure Electronic Transaction (SET)

As the Internet became more popular and e-commerce became the name of the game, the payment card industry (PCI)—that is, the banks that issue credit cards—recognized the need to protect credit card information when it is used for Internet transactions. Their first attempt to protect this information was called *Secure Electronic Transaction (SET)*.

When a client browsed to an e-commerce website and wanted to make a credit card purchase, SET, using X.509 digital certificates, would validate identities and encrypt the credit card information. That encrypted credit card data was sent through the vendor (the e-commerce website) to the vendor's bank, called the merchant bank. The credit card data would be decrypted at the merchant bank, where the issuer and the client's information would be revealed. This kept the credit card data encrypted and secure in transit from the client computer, across the Internet, to the vendor's bank, a trusted endpoint. The vendor never has access to the plaintext credit card numbers, so the credit card data can never be exposed by the vendor. The vendor's bank would then contact the issuing bank for approval of the charge. If the issuing bank approved the charge (its promise to pay), the vendor's bank would notify the e-commerce website to accept the order. The settlement of payments would typically occur within three days of the approval, and the vendor's bank account would receive funds from the issuing bank. The issuing bank would then bill the client on the pre-established monthly billing cycle (see Figure 3-46).

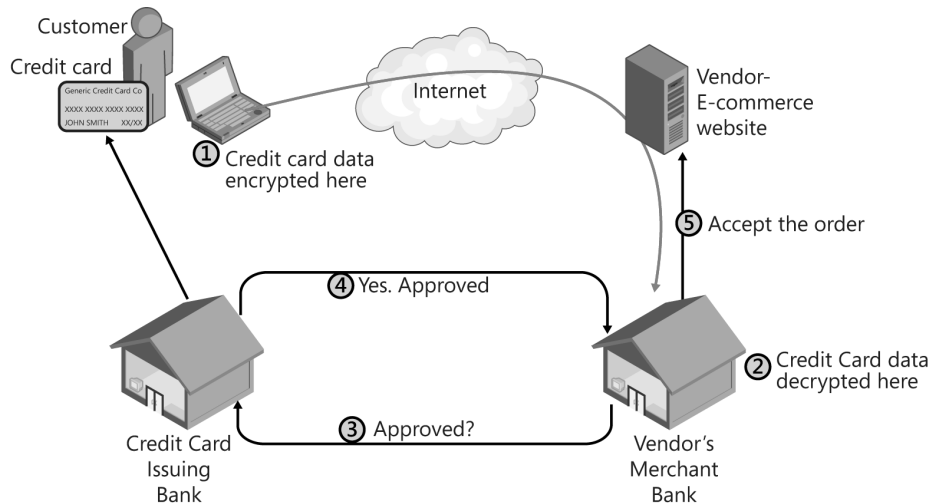


FIGURE 3-46 Secure electronic transaction (SET)

This sounds wonderful until the vendors learned that the banks wanted to charge them a transaction fee of a small percentage of the transaction plus another small percentage for the use of SET. Vendors refused to pay this second fee to help protect the banks, and SET never had a chance of survival.

The banks then decided that if the vendors would not pay the SET fees, the vendors must provide adequate security for the credit card data and implemented the Payment Card Industry Data Security Standard (PCI-DSS). This obligates all vendors who accept credit cards to meet and verify information system security standards designed to protect the credit card data.

Secure Multipurpose Internet Message Extensions (S/MIME)

Earlier in this chapter, you saw how email messages can be secured by using digital certificates for signing and sealing. What about the email attachments? Multipurpose Internet Message Extensions (MIME) is the Layer 6 protocol that enables the Simple Mail Transfer Protocol (SMTP), the protocol that is used to send email and include file attachments with the email messages. By default, MIME does not provide any security protections. Secure MIME (S/MIME), developed by RSA, uses X.509 digital certificates, and the S/MIME feature has been built into virtually every email system to encrypt and digitally sign the attachments of protected email messages.

Steganography

In a category within cryptography, somewhat of its own, is steganography. The term *steganography* is from Greek, meaning “covered writing.” It was recognized that data can be relatively easily hidden within certain types of files. Digital audio, video, and still photographs provide this opportunity. On a typical digital camera today, photographs contain 10 megapixels or

more. That is approximately 10,000,000 color dots that collectively make the image. Each pixel has six attributes:

- Red
- Green
- Blue
- Hue
- Saturation
- Luminance

Each of these attributes is described by at least eight binary bits, providing 256 shades of red, from no red (00000000) to maximum red (11111111), 256 shades of green, and so on. (Many cameras today provide 32-bit color, which equates to about four billion shades of red and green and blue and so on.) The human eye cannot detect the subtle change from one level of red to the next $1/256$ level of red, so binary data can be written as the least significant bit of each attribute (six bits per pixel) and across the entire picture of 10,000,000 pixels. This simple algorithm would allow you to hide approximately 7.5 megabytes ($(6 \text{ bits per pixel} \times 10,000,000 \text{ pixels})/8 \text{ bits per byte} = 7.5 \text{ MB}$) of data in a single photograph without visibly altering the photograph. Furthermore, consider that many of the original least-significant bits will already be the bits required by the hidden data, causing even less visual perturbation to the photo.

Although the steganography algorithms are more complex, the concepts remain the same. Hide your data invisibly within these color dots (pixels), or audio clips, or video clips. Only you and your counterpart know there is data to be found and how to find it. This awareness of the hidden data is the nature of the symmetric key in steganography. For greater protection, you could encrypt the data before encoding it into an image file. In this case, steganography becomes a protected, covert storage channel.

Steganography provides a different service than most encryption technologies—secrecy instead of confidentiality. If a sender is concerned that an attacker could intercept and compromise a sensitive message during transmission to a remote recipient, the sender can apply an encryption process (cryptography) to the message and then transmit the ciphertext securely. Although an attacker might see the ciphertext transmission, he can't read or understand the encrypted message, but he is fully aware that a sensitive message has just been transmitted. Cryptoprocesses provide confidentiality services—C for crypto and C for confidentiality.

However, if the sender chose to use steganography instead of cryptography, the sensitive message could be encoded in an image file (or any of the other file types that support good steganography); an attacker would see an uninteresting image file and would be unaware that a hidden message exists or was communicated. This is secrecy of information flow, not confidentiality. Stego provides secrecy services—S for stego and S for secrecy.

To take this process a step further, the sender could first encrypt the sensitive message to provide confidentiality (just in case the message is discovered) and then use steganography

to provide secrecy, embedding the ciphertext message in an appropriate carrier file. Combining these two technologies provides secrecy and confidentiality. Cool.

Watermarks

Steganography can be used for another purpose as well. To identify proprietary image, audio, and video digital content, steganography can be used to hide an invisible digital watermark. Then active processes can search web content for these watermarks that are acting as a hidden fingerprint identifying the true owner of the content. These active processes are referred to as spiders, crawlers, robots, and sometimes just bots. If a spider locates one of its watermarks, the spider reports back to the owner, who would verify whether proper licensing fees have been paid for the use of the content.

Watermarks might also be visible to act as directive control labeling the content for a certain type of use or protective control, as an advertisement to steer sales to the legitimate owner, or to act as a deterrent for illegitimate use.

Attacks on cryptography

The following are classes of attacks on cryptography. They describe how much information and access the attacker has while actively attacking the cryptosystem. The attacker's goal is to crack (reveal) the encryption keys, decryption keys, or both.

Ciphertext-only attack

When all is designed and implemented in its best forms, a cryptosystem shows only one thing to the unauthorized bad guys: ciphertext. The bad guys have no information about the plaintext data or the nature of the keys the cryptosystem uses. This is the good guy's strongest position and the bad guy's weakest position. Considering that the bad guy wants to steal your messages and crack your keys, put the bad guy in this position whenever you can.

Known plaintext attack

If the bad guy knows what plaintext was entered to produce the resulting ciphertext, he gains insight into the nature of the encryption keys used to create the ciphertext. It is this type of insight that reduces the work factor of the cryptosystem, weakening the security in the environment.

When an attacker knows the plaintext and sees the resulting ciphertext, the attack on the keys becomes more direct. The attacker's position just got a little stronger, and your position just got a little weaker. If a bad guy had captured some ciphertext and could learn something about the plaintext content that was part of the message, she could more easily crack the encryption key.

Chosen plaintext attack

If the bad guy could choose some plaintext to be encrypted using the victim's key, he could choose patterns of plaintext that might more likely lead to patterns within the resulting ciphertext. Suppose that a coworker tends to take long breaks and often fails to lock her workstation. An attacker could use the unlocked system to generate plaintext messages at will and then use the victim's encryption key to produce ciphertext. Another example is the use of someone's public key to encrypt any content desired and then inspect the resulting ciphertext, hoping to gain insight into the private key of the victim.

Symmetric key cryptosystems present a vulnerability in this area. If an attacker chooses a plaintext message that is the same size as the key length and is all zeros and then encrypts the message with the victim's system, the resulting ciphertext will be the victim's symmetric key, as demonstrated in Figure 3-47.

Chosen Plaintext	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Key	1	1	0	0	0	1	1	1	0	1	0	1	1	0	1
Ciphertext	1	1	0	0	0	1	1	1	0	1	0	1	1	0	1

FIGURE 3-47 Chosen plaintext attack on symmetric keys

The chosen plaintext attack is useless on asymmetric key ciphertext messages because these are encrypted by using the potential victim's public key that is already available to the attacker. The attack does not in any way expose the potential victim's private key.

Chosen ciphertext attack

This attack moves in the opposite direction. In the chosen ciphertext attack, the attacker creates ciphertext, usually some patterns of characters. Then he runs the ciphertext through the victim's system to reveal the decrypted plaintext. This direction produces a stronger link to the nature of the decryption key of the victim. Cryptosystems are designed to produce strong ciphertext by hiding (randomizing) any nature of the key within the resulting ciphertext. The concern was never to hide the nature of the key in the decrypted plaintext.

This attack puts the good guy in her weakest position and the attacker in his strongest position.

Adaptive attacks

The adaptive attack is a variation or an extension of either the chosen plaintext or the chosen ciphertext attack. By gaining some initial insight to the nature of the key, an attacker might adjust the attack vector and attack again, gaining some additional insight and adjusting her attack vector to improve her position once again. These can also be called iterative chosen plaintext and iterative chosen ciphertext attacks.

Exercises

In the following exercises, you apply what you've learned in this chapter. The answers for these exercises are located in the "Answers" section at the end of this chapter.

Exercise 3-1

Download a cryptography utility from a trusted source. An example might be TrueCrypt from <http://www.truecrypt.org>. Install the application and establish an encrypted volume. Notice the cryptographic functions required to build and mount the encrypted volume, including the encryption algorithm, password creation, and key creation.

Identify the cryptographic ciphers and the hashing algorithms implemented by the application and the type of cipher.

Exercise 3-2

Locate the Trusted Root Certification Authorities list on your computer. In Internet Explorer, this can be found by choosing Tools | Internet Options | Content | Certificates | Trusted Root Certification Authorities. Open and examine several root certificates. On the public CA's website, locate and review the certificate practices statement (CPS) for one or more of the public root CAs.

Identify the algorithms used for the following attributes in three of the digital certificates stored in the Trusted Root Certification Authorities store on your computer:

- Signature cryptographic algorithm
- Signature hashing algorithm
- Public key algorithm
- Public key length
- Thumbprint (hashing) algorithm

Chapter summary

- Cryptanalysis is the science of cracking cryptosystems. It might be done for good cause, to prove the strength of a cryptosystem, or for bad cause, to steal secret messages and keys.
- Hashing algorithms produce a fingerprint of a message and can be used to provide integrity validation and authentication.
- Contemporary symmetric key cryptography is fast and includes two types of algorithms: block ciphers and stream ciphers.

- Block ciphers and stream ciphers might use initialization vectors (IVs) to randomize and strengthen the resulting ciphertext. The IV is a nonsecret value and might also be called a salt, a seed, or a nonce.
- Block ciphers tend to show patterns and might require the use of randomizing modes. The five prevalent modes of symmetric key block ciphers are: Electronic Code Book (ECB); cipher block chaining (CBC); Output Feedback mode (OFB); Cipher Feedback mode (CFB); and Counter mode (CTR).
- Symmetric key cryptography causes the need to distribute symmetric keys securely but can provide three cryptographic services: confidentiality (strong), authentication (weak); and Integrity validation (weak).
- Hybrid cryptosystems use combinations of symmetric key algorithms, asymmetric key algorithms, and hashing algorithms and can provide all five cryptographic services in a strong manner. They include confidentiality, authentication, nonrepudiation, integrity validation, and secure key distribution.
- The first asymmetric algorithm, the Diffie-Hellman-Merkle algorithm, was introduced in 1976. It is based on calculating discrete logarithms in a finite field. It was designed to solve only one of the five cryptographic services: secure key distribution.
- A public-key infrastructure (PKI) is based on a hierarchical trust model of certification authorities (CAs). It uses X.509 digital certificates to bind a public key to a user's identity.

Chapter review

Test your knowledge of the information in this chapter by answering these questions. The answers to these questions, and the explanations of why each answer choice is correct or incorrect, are located in the "Answers" section at the end of this chapter.

1. Which feature or mode of IPsec protects the actual source and destination IP addresses?
 - A. Transport mode
 - B. Tunnel mode
 - C. Internet Key Exchange (IKE)
 - D. Security Parameter Index (SPI)
2. What network location needs to be checked to verify whether the issuer of an X.509 digital certificate has withdrawn its claim of trust for the subject of the certificate?
 - A. The certificate revocation list (CRL)
 - B. Directory services registry
 - C. The email server for the organization
 - D. The CRL distribution point (CDP)

3. Which of the following is not a characteristic of the one-time pad?
 - A. The key must be as long as the message.
 - B. The key is never reused.
 - C. The key is bound to a certificate.
 - D. The key should be highly randomized.
4. Which of the following describes an attacker attempting to create a collision by using a hashing algorithm?
 - A. The brute force attack
 - B. The rainbow attack
 - C. The birthday attack
 - D. The ciphertext-only attack
5. Which of the following accurately describes the requirements to produce a signed and sealed message?
 - A. Sender's public key, sender's private key, and a hashing algorithm
 - B. Sender's private key, recipient's public key, and a hashing algorithm
 - C. Sender's private key, recipient's private key, and a hashing algorithm
 - D. Sender's private key, recipient's public key, and recipient's private key
6. Which of the following accurately describes the attack used to reveal the sender's symmetric key easily?
 - A. Ciphertext only
 - B. Known plaintext
 - C. Chosen plaintext
 - D. Chosen ciphertext
7. Which of the following is the fastest and strongest mode of symmetric key block ciphers?
 - A. Counter mode
 - B. Output Feedback mode
 - C. Cipher Feedback mode
 - D. Cipher block chaining
8. Which function is used in the S-box in contemporary symmetric key cryptosystems?
 - A. Transposition
 - B. Hashing
 - C. Pseudo-random number generation
 - D. Exclusive Or

Answers

This section contains the answers to the exercises and the “Chapter review” section in this chapter.

Exercise 3-1

Following is the list of cryptographic ciphers implemented by the TrueCrypt v7.1 application:

- AES
- Serpent
- Twofish
- Combination of these

Following is the list of hashing algorithms implemented by the TrueCrypt v7.1 application:

- RIPEMD-160
- SHA512
- Whirlpool

NOTE: A different version of the TrueCrypt application might implement different algorithms.

Exercise 3-2

Following are examples of three certificates that might be found in the Trusted Root Certification Authorities store on a computer system.

- Entrust.net Certification Authority (2048):
 - Signature cryptographic algorithm: RSA
 - Signature hashing algorithm: SHA1
 - Public key algorithm: RSA
 - Public key length: 2048 bits
 - Thumbprint (hashing) algorithm: SHA1
- Equifax Secure Certificate Authority:
 - Signature cryptographic algorithm: RSA
 - Signature hashing algorithm: SHA1
 - Public key algorithm: RSA
 - Public key length: 1024 bits
 - Thumbprint (hashing) algorithm: SHA1
- Microsoft Root Certification Authority 2011:
 - Signature cryptographic algorithm: RSA
 - Signature hashing algorithm: SHA256

- Public key algorithm: RSA
- Public key length: 4096 bits
- Thumbprint (hashing) algorithm: SHA1

Chapter review

1. Correct answer: B

- A. Incorrect:** Transport mode presents the actual source and destination IP addresses in plaintext.
- B. Correct:** Tunnel mode encrypts the actual source and destination IP addresses and is added to the beginning of the packet with a new Layer 3 header.
- C. Incorrect:** IKE is used to negotiate the terms of the IPsec VPN securely.
- D. Incorrect:** The SPI is used to look up the processing details quickly for each IPsec packet.

2. Correct answer: D

- A. Incorrect:** The CRL is the list of revoked certificates. It is not a network location.
- B. Incorrect:** The directory services registry is used to record details for directory services. It does not contain a list of revoked certificates.
- C. Incorrect:** An email server might be used as a certificate repository but not a CRL distribution point.
- D. Correct:** The CDP is a network location where the CRL is published.

3. Correct answer: C

- A. Incorrect:** The one-time pad requires a key that is as long as the message.
- B. Incorrect:** A key is never reused in a one-time pad.
- C. Correct:** The public key in a PKI is bound to a digital certificate. This has nothing to do with a one-time pad.
- D. Incorrect:** The one-time pad avoids patterns in the ciphertext by using highly randomized key values.

4. Correct answer: C

- A. Incorrect:** The brute-force attack is an exhaustive attack on cryptographic keys.
- B. Incorrect:** The rainbow attack uses rainbow tables to reverse-lookup passwords from a password store that hashes the passwords.
- C. Correct:** The birthday attack is an attempt to produce the same hash value for a modified message.
- D. Incorrect:** In the ciphertext-only attack, the attacker only sees ciphertext. This makes the attacker's job more difficult.

5. Correct answer: B

- A. Incorrect:** The digital signature requires the sender's private key and a hashing algorithm. The sealing of a message requires the recipient's public key.
- B. Correct:** The digital signature requires the sender's private key and a hashing algorithm. The sealing of a message requires the recipient's public key.
- C. Incorrect:** The digital signature requires the sender's private key and a hashing algorithm. The sealing of a message requires the recipient's public key.
- D. Incorrect:** The digital signature requires the sender's private key and a hashing algorithm. The sealing of a message requires the recipient's public key. The recipient's private key is only needed to decrypt the message, not to produce the message.

6. Correct answer: C

- A. Incorrect:** The ciphertext-only attack is the most difficult of all attacks.
- B. Incorrect:** The known plaintext attack allows the attacker to compare the plaintext to the resulting ciphertext and is used to perform the meet-in-the-middle attack, like that used on 2DES.
- C. Correct:** By choosing a block of all zeroes as the chosen plaintext, the results should show the symmetric key used to encrypt the messages.
- D. Incorrect:** The chosen ciphertext attack is used to reveal the recipient's asymmetric private key.

7. Correct answer: A

- A. Correct:** Counter mode (CTR) is the fastest and strongest mode of symmetric key block ciphers.
- B. Incorrect:** Output Feedback (OFB) is faster than CFB but slower than CTR, ECB, and CBC modes of symmetric key block ciphers.
- C. Incorrect:** Cipher Feedback (CFB) is slower than CTR, ECB, CBC, and OFB modes of symmetric key block ciphers.
- D. Incorrect:** Cipher block chaining (CBC) is faster than OFB and CFB but slower than CTR and ECB modes of symmetric key block ciphers.

8. Correct answer: D

- A. Incorrect:** Transposition mixes the order of the characters of the message.
- B. Incorrect:** Hashing produces a fingerprint of a message.
- C. Incorrect:** The PRNG is used in most stream ciphers to help produce a keystream.
- D. Correct:** The Exclusive Or (XOR) function is used inside the S-boxes in symmetric key block ciphers. The S is for substitution.

Index

Symbols

- 2DES (double DES), 177
- 3DES (triple DES), 177–178
- 4G LTE (Long Term Evolution) vs. 4G cellular, 119
- 10BASE5 coax (RG-8/U, Thicknet), 446
- 802.11i enterprise authentication, 501–503
- 802.11n and 802.11ac, MIMO data streams in, 503
- ® symbol for registered and approved trademarks, 375

A

- AAA (Authentication Authorization Auditing) functions
 - about, 70
 - auditing
 - about, 120–124
 - honeypots, honeynets and padded cells in, 129
 - intrusion detection and prevention systems, 124–129
 - authentication
 - Kerberos, 94–100
 - mutual, 90–93
 - Sesame, 100–101
 - web-based, 101–102
 - authentication categories
 - someplace you are, 89
 - something you are, 84–89
 - something you have, 81–84
 - something you know, 77–81
 - authorization
 - about, 103–104
 - centralized access control, 115–119
 - constrained interface, 119
 - decentralized access control, 115
 - discretionary access control, 109–112
 - hardware guard, 119
 - hybrid access control, 115
 - life cycle, 104–105
 - mandatory access control, 105–109
 - role-based access control, 113–114
 - rule-based access control, 114–115
 - software guard, 119
 - temporal access controls, 119
 - identity management and, 76
 - multi-factor (two-factor) authentication, 89–90
 - process of managing access using, 74
 - services
 - using Diameter, 116, 118–119
 - using RADIUS, 116
 - using TACACS, 117
 - single sign on, 93–94
- A bit (archive bit), 684
- ABM (Asynchronous Balanced Mode), 463
- ABRs (Specified Area Border Routers), 482
- acceptable use policy, informing prospective employees of, 46–47
- acceptance testing, in software development, 581, 590
- accepting risk, countermeasure of, 22–23, 33
- Access Control List (ACL), 110, 112
- access control matrix, in DAC, 112
- access controls. *See also* permissions
 - administrative
 - about, 11, 66, 70
 - MAC I and, 108
 - assessing effectiveness of, 71
 - assessment, 68
 - centralized, 115–119
 - compensating, 69
 - compensating controls and, 104
 - constrained interface, 119
 - corrective, 68–69
 - countermeasures and, 10–12
 - DAC as identity-based, 110

Access Points (APs), wireless

- decentralized, 115
- delay, 67
- detective, 67–68
- deterrent, 67
- directive, 69
- embodying functional security objectives, 70
- hardware guard, 119
- hybrid, 115
- MAC model, 323–326
 - physical
 - about, 11, 66, 70
 - MAC I and, 108
 - preventive, 67
 - protective controls for, 42
 - role-based, 113–114
 - software guard, 119
 - technical
 - about, 11, 66, 70
 - DAC as, 109–112
 - MAC I and, 105, 108
 - RBAC, 323
 - role-based access control as, 113
 - rule-based access control, 114–115
 - temporal, 119–120
 - vs. countermeasures, 65
- Access Points (APs), wireless, 493–494
- access, process of managing, 74
- access statement, in trusted path, 64
- access token, KDC login and TGT as user, 96
- access triple, 342
- accountability, 76
- account creation, 71
- accounting, 76
- account lockout feature
 - password, 77
 - PIN, 78
- accreditation and certification, of policy documents, 19–20
- ACFE (Association of Certified Fraud Examiners), on enterprise losses due to fraud, 657
- ACK (Acknowledgment), TCP segment, 426, 427–428
- ACL (Access Control List), 110, 112
- acoustic sensors, in IDS, 273
- Active Directory
 - domain controller, 95
 - X.500 Directory Services and, 99
- active reconnaissance, in targeted attacks, 695–696
- ActiveX controls, 604
- ActiveX Data Objects (ADO), 611
- adaptive attacks, 237
- address bus, computer system, 311
- Address Resolution Protocol (ARP), 437, 438, 479, 509
- AddRoundKey function, 178
- add to in-person authentication in PKI system, 214
- add to that biometric authentication mechanism in PKI system, 214
- Add to that DNA verification in PKI system, 214
- Adepto, 402
- ad hoc mode, wireless network, 495
- Adleman, Leonard, 205
- administrative access controls
 - about, 11, 66
 - functional security objectives and, 70
 - MAC I and, 108
 - nondisclosure agreement as, 104
- administrative controls, fraud protection, 658–659
- administrative law, 372
- admissibility of evidence, 397–398
- ADO (ActiveX Data Objects), 611
- Advanced Encryption Standard (AES)
 - as symmetric key block cipher, 178–179
 - crypto functions, 178
 - historical review of, 157
- Advanced Persistent Threats (APTs), 366, 369, 693
- Advanced Research Projects Agency Network (ARPANET), 441
- adware, 630
- AES (Advanced Encryption Standard)
 - as symmetric key block cipher, 178–179
 - crypto functions, 178
 - historical review of, 157
- aggregation and inference attacks, 340–341, 613
- AH (Authentication Header), 226–227
- AI (Artificial Intelligence), 376, 390, 620–624
- air gapping (network isolation), as compensating control, 69
- ALE (Annual Loss Expectancy)
 - calculation, 29–31
 - countermeasures reducing, 31–32
- algorithms. *See also* ciphers
 - about, 142–143
 - artificial intelligence, 390
 - asymmetric key. *See* asymmetric key algorithms
 - components of strong, 168–169
 - Diffie-Hellman, 156, 190, 202–205

- DSA, 206–207
- DSS, 206–207
- ECC, 205–206
- ElGamal, 206
- hashing. *See* hashing algorithms
- Knapsack, 207
- LUC, 207
- Lucifer, 156
- Rijndael, 157
- RSA asymmetric key, 205
- symmetric key. *See* symmetric key algorithms
- XTR, 207
- Allow access permissions, 74
- allow permissions, applied to No Access default, 103
- ALU (Arithmetic Logic Unit), 307
- AM (Amplitude Modulation), 450
- American DataBank
 - study on employee lawsuits and workplace crime, 45
 - study on rejection of employee candidates, 46
- Amplitude Modulation (AM), 450
- analog encoding, 450–452
- analysis
 - forensic, 400–401
 - in incident response system, 394
- ANN (Artificial Neural Network), 624
- Annualized Rate of Occurrence (ARO), 29
- Annual Loss Expectancy (ALE)
 - calculation, 29–31
 - countermeasures reducing, 31–32
- anomaly-based (behavior-based) detection mechanism, 126, 638
- Antheil, George, 495
- antivirus (AV)
 - signatures, 591
 - software
 - attackers evading, 699
 - monitoring output from, 390
- Anycast mode, 458
- APIs (Application Programming Interfaces)
 - designing, 315–316
 - tools to define, 590
- AppArmor operating system, MAC implementation in, 107
- application architectures, 326–332
- application-independent technology, 601
- Application layer of OSI model, 421–422
- Application Programming Interfaces (APIs)
 - designing, 315–316
 - tools to define, 590
- applications, attacks on
 - buffer overflow attack, 320–321, 625
 - cookies, 637
 - covert communications channels, 627
 - directory transversal attacks, 636
 - failure to release memory securely, 626
 - malware
 - about, 628–631
 - detection mechanisms, 637–638
 - race conditions, 627–628
 - residual maintenance hooks, 626
 - sensitive data retrieval, 636
 - SQL injection attack, 625
 - web-based applications, 632–634
 - web cache poisoning, 634–635
- APs (Access Points), wireless, 493–494
- APTs (Advanced Persistent Threats), 366, 369, 693
- arbitrary substitution cipher, 151
- ARCFOUR (ARC4), 175
- archive bit (A bit), 684
- ARCnet (Attached Resource Computer Network), 454, 486
- Arithmetic Logic Unit (ALU), 307
- Army, US, using cryptography, 154
- ARO (Annualized Rate of Occurrence), 29
- ARP (Address Resolution Protocol), 437, 438, 479, 509
- ARPANET (Advanced Research Projects Agency Network), 441
- Artificial Intelligence (AI), 376, 390, 620–624
- Artificial Neural Network (ANN), 624
- AS (Authentication Service)
 - about, 75
 - asynchronous token devices using, 83
 - Kerberos support of, 94–95
 - Kerberos using, 96
- AS (Autonomous Systems), 458
- assessment controls, 68
- assessment of severity of intrusion, 250, 251–252
- assets
 - identifying exposures, 10
 - information
 - assigning value to, 43
 - definition of, 5
 - inventorying, 43
 - reappraising and adjusting classification of, 44
 - vulnerabilities of, 8, 9

asset tracking, as part of securing portable devices

- in performing risk assessment
 - assigning values to, 25–28
 - classifying, 28
 - intangible, 25
 - inventory, 24–25
 - tangible, 24–25
- reviewing list of company-owned physical, 48
- vulnerabilities of systems, 8, 9
- asset tracking, as part of securing portable devices, 271
- Asset Value (AV)
 - about, 28
 - in SLE, 29
- assigned privileges, 104
- assisted password reset, 79
- Association of Certified Fraud Examiners (ACFE), on
 - enterprise loses due to fraud, 657
- assurance, in computer system security, 344
- asymmetric key algorithms
 - about, 170, 190–191
 - length of keys, 191
 - list of, 201–207
 - mathematical keys in, 202
 - sealing messages using, 195–201
 - signing messages using, 192–195, 198–201
- asymmetric key cryptography
 - ciphers, 148
 - ITU-T and, 100
 - keys in
 - distribution of, 161
 - length of, 160–161
 - management of, 159, 191
 - mathematical, 202
 - public vs. private key, 191
 - quantities of, 162–163
 - providing nonrepudiation, 140
 - SESAME using, 100
- asymmetric public key, 83
- Asynchronous Balanced Mode (ABM), 463
- asynchronous token devices, to authenticate users, 83
- Asynchronous Transfer Mode (ATM)
 - about, 490
 - L2TP network, 229
 - supporting pinned path requirements, 380
- asynchronous transmissions, 453
- Atbash cipher, 149
- ATM (Asynchronous Transfer Mode)
 - about, 490
 - L2TP network, 229
 - supporting pinned path requirements, 380
- ATMs (Automatic Teller Machines), constrained interface
 - in, 119
- atomic
 - definition of, 617
 - transactions, 616
- Attached Resource Computer Network (ARCnet), 486
- attackers
 - entrapment of, 130
 - logging system protections against, 123
 - using IPS, 129
- attacker system, computer in cybercrime as, 368
- attack surface (exposure), identifying, 10
- attack vector, ciphertext as, 165
- attenuated signal, 444
- attenuation, 261
- auditing
 - about, 51, 75–76, 120–124
 - architecture, 127
 - DAC supporting, 111
 - detering fraud using, 73
 - for protection of information, 379–382
 - honeypots, honeynets and padded cells in, 129
 - internal, 122
 - intrusion detection systems and
 - about, 124
 - components of, 126
 - detection mechanisms used by, 125–126
 - network-based and host-based, 125
 - response system, 125
 - vs. intrusion prevention systems and, 125
 - intrusion prevention systems and
 - about, 124
 - architecture, 127
 - attackers using, 129
 - detection mechanisms used by, 125–126
 - mechanisms to defeat sensors, 127
 - vs. intrusion detection systems, 125
 - monitoring and, 51–54
 - systems for logging and, 120–124, 280
 - user activity, 71
- auditor role, in classifying data, 40
- authentication
 - in cryptosystem, 143
 - Kerberos, 94
 - mutual, 90–93
 - protocols, 484
 - provided by digital signatures, 192
 - providing claim of identity, 140

- Sesame, 100–101
- symmetric keys and, 170–171
- techniques providing MAC, 185–189
- web-based, 101–102
- Authentication Authorization Auditing (AAA) functions
 - about, 70
 - auditing
 - about, 120–124
 - honeypots, honeynets and padded cells in, 129
 - intrusion detection and prevention systems, 124–129
 - authentication
 - Kerberos, 94–100
 - mutual, 90–93
 - Sesame, 100–101
 - web-based, 101–102
 - authentication categories
 - someplace you are, 89
 - something you are, 84–89
 - something you have, 81–84
 - something you know, 77–81
 - authorization
 - about, 103–104
 - centralized access control, 115–119
 - constrained interface, 119
 - decentralized access control, 115
 - discretionary access control, 109–112
 - hardware guard, 119
 - hybrid access control, 115
 - life cycle, 104–105
 - mandatory access control, 105–109
 - role-based access control, 113–114
 - rule-based access control, 114–115
 - software guard, 119
 - temporal access controls, 119
 - identity management and, 76
 - multi-factor (two-factor) authentication, 89–90
 - process of managing access using, 74
 - services
 - using Diameter, 116, 118–119
 - using RADIUS, 116
 - using TACACS, 117
 - single sign on, 93–94
- authentication categories
 - someplace you are, 89
 - something you are, 84–89
 - something you have, 81–84
 - something you know, 77–81
- Authentication Header (AH), 226–227
- authentication process, identification and, 75
- Authentication Service (AS)
 - about, 75
 - asynchronous token devices using, 83
 - Kerberos support of, 94–95
 - Kerberos using, 96
- authorization
 - about, 75, 103–104
 - centralized access control, 115–119
 - constrained interface, 119
 - decentralized access control, 115
 - discretionary access control, 109–112
 - hardware guard, 119
 - hybrid access control, 115
 - Kerberos authentication vs., 98
 - life cycle, 104–105
 - mandatory access control, 105–109
 - role-based access control, 113–114
 - rule-based access control, 114–115
 - software guard, 119
 - temporal access controls, 119
- authorization creep (authorization aggregation), 71, 656
- authorized personnel, protective controls for, 41
- automated recovery, 559
- Automatic Teller Machines (ATMs), constrained interface
 - in, 119
- Autonomous Systems (AS), 458
- availability
 - breaches of, 7
 - definition of, 6–7
 - losses and, 8
 - of operation systems, 655–656
- AV (antivirus)
 - signatures, 591
 - software
 - attackers evading, 699
 - monitoring output from, 390
- AV (Asset Value)
 - about, 28
 - in SLE, 29
- avoiding risk, countermeasure of, 22–23, 33

B

- backdoor attack, information theft by, 510, 630–631
- back-end application servers, 326
- background checks, for prospective employees, 46
- backups
 - batch, 552
 - data, 683–687
 - real-time, 552
 - scheduling, 683
 - security requirements for data, 553
 - storage location of data, 553
 - storage of, 558–559
 - strategies for, 555–558
- Banyon VINES operating system, X.500 Directory Services and, 99
- base address, logical address as, 321
- baseband vs. broadband transmissions, 446
- baseline documents, 16
- Basic Input/Output System (BIOS), 306
- basic level authentication in PKI system, 214
- bastion host/hardened systems, 465–467
- bastion host system, KDC installed on, 95
- batch backups, 552
- batch file, writing as SSO solution, 94
- BCP (Business Continuity Plan)
 - about, 4, 528–529
 - developing plan proposals
 - about, 540
 - alternative leased sites, 545–546
 - backup strategies and storage, 555–559
 - collocation of processes, 544
 - considering alternative procedures, 542
 - identifying preventive controls, 541
 - reciprocal agreements, 546–547
 - recognizing increased operating costs, 542
 - recovery of data, 551–554
 - recovery of personnel, 559–560
 - recovery of supply systems, 547–548
 - recovery of technologies, 548–550
 - rolling hot sites, 546
 - security standards, 550
 - workspace recovery, 543–544
 - developing reconstitution guidelines, 560–561
 - development of, 525–526
 - identifying single points of failure, 655
 - implementing approved plans, 562–563
 - presentation to senior management, 561–562
 - sharing accomplishment, 570
 - stages of planning process
 - about, 529–530
 - defining need, 530
 - defining planning budget and schedule, 532–533
 - defining planning project leader, 530
 - defining planning scope of project, 531
 - defining planning team, 531
 - performing business impact analysis, 533–540
 - timeline for, 529
- behavior-based (anomaly-based) detection mechanism, 126, 638
- Bell-LaPadula (BL) model
 - computer systems, 339
 - Orange Book based on, 745
- Berkeley Internet Name Domain (BIND) daemon, 473
- best evidence, 398
- beyond a reasonable doubt, burden of proof, 372
- BGP (Border Gateway Protocol), 482
- bias, in surveys, 28
- Biba model, computer system, 340–342
- BIND (Berkeley Internet Name Domain) daemon, 473
- biometric systems, authentication using
 - about, 84
 - devices for, 313
 - drawbacks of, 88–89
 - enrollment process for, 85
 - errors in, 85–86
 - finding matching record, 87–89
 - techniques used in, 84–85
- BIOS (Basic Input/Output System), 306
- bipolar signaling, 452–453
- BitLocker Drive Encryption (Microsoft) tool, 141
- black box testing, 667
- Blacker operating system, MAC implementation in, 107
- blackout, electrical, 278
- BL (Bell-LaPadula) model
 - computer systems, 339
 - ITSEC based on, 347–348
 - Orange Book based on, 345
- block ciphers
 - AES as symmetric key, 157, 178–179
 - Blowfish as symmetric key, 179
 - CBC-MAC, 187–188
 - double DES as symmetric key, 177
 - Feistel Network and symmetric key, 159
 - IDEA as symmetric key, 177

- Lucifer as symmetric key, 156
 - modes of symmetric key, 180–184
 - RC4 as symmetric key, 175
 - RC5 as symmetric key, 179
 - RC6 as symmetric key, 179
 - symmetric key, 173, 175–179
 - triple DES as symmetric key, 177
 - Twofish as symmetric key, 179
 - block-level storage of data, 680
 - blocks, hard disk, 403–404
 - Blowfish, as symmetric key block cipher, 179
 - bluejacking wireless networks, 512
 - Bluetooth, 486
 - BN (Brewer-Nash) model, computer system, 343
 - bogon, 635
 - bollards, in designing physical security, 261
 - Boolean logic, 172–173
 - Boole, George, 158
 - bootstrap operating system, 306
 - Bootstrap Protocol (BootP), 475, 479
 - Border Gateway Protocol (BGP), 482
 - boundaries, identifying architectural, 304–305
 - Brewer-Nash (BN) model, computer system, 343
 - bridges, 461–462
 - Bring Your Own Device (BYOD), 331
 - Bring Your Own Device (BYOD) client systems, 464–465
 - British Standard 7799 (BS7799), 333
 - broadband vs. baseband transmissions, 446
 - Broadcast mode, 458
 - brownout, electrical, 278
 - brute force attack
 - cracking cryptosystem with, 144
 - on passwords, 79
 - BSA (Business Software Alliance), 384
 - Budapest Convention, 384
 - buffer memory, operating system, 318, 319, 322
 - buffer overflow attack, 320–321, 625
 - building materials of facility, as physical control in designing physical security, 255–256
 - burn rating, as physical control in designing physical security, 255
 - bus, computer system, 311
 - Business Continuity Plan (BCP)
 - about, 4, 528–529
 - components of plans, 563–569
 - developing plan proposals
 - about, 540
 - alternative leased sites, 545–546
 - backup strategies and storage, 555–559
 - collocation of processes, 544
 - compliance with laws and regulations, 542
 - considering alternative procedures, 542
 - identifying preventive controls, 541
 - reciprocal agreements, 546–547
 - recognizing increased operating costs, 542
 - recovery of data, 551–554
 - recovery of personnel, 559–560
 - recovery of supply systems, 547–548
 - recovery of technologies, 548–550
 - rolling hot sites, 546
 - security standards, 550
 - workspace recovery, 543–544
 - developing reconstitution guidelines, 560–561
 - development of, 525–526
 - identifying single points of failure, 655
 - implementing approved plans, 562–563
 - presentation to senior management, 561–562
 - sharing accomplishment, 570
 - stages of planning process
 - about, 529–530
 - defining need, 530
 - defining planning budget and schedule, 532–533
 - defining planning project leader, 530
 - defining planning scope of project, 531
 - defining planning team, 531
 - performing business impact analysis, 533–540
 - timeline for, 529
 - business functions, cyclical nature of, 539
 - business impact analysis
 - evaluation of MTD, 538
 - performing, 533–540
 - Business Software Alliance (BSA), 384
 - BYOD (Bring Your Own Device), 331
- ## C
- cable locks, securing portable devices with, 270
 - cable modem, 488
 - cables
 - in designing physical security, 262–263
 - types of, 445–449
 - CAC (Common Access Card) smart card, private key encrypted on, 217

C&A (Certification and Accreditation) frameworks

- C&A (Certification and Accreditation) frameworks, 344–349
- cache memory, computer system, 310
- Caesar (Shift) cipher, 150–151
- camera security, in designing physical security, 267–270
- Campus Area Networks (CANs), 488
- Candidate Information Bulletin (CIB), 383
- CANs (Campus Area Networks), 488
- capability, in access control matrix, 112
- Capability Maturity Model Integration (CMMI), five stages of, 587–588
- Carnegie Mellon University (CMU), 386
- Carrier Sense Multiple Access with Collision Detection (CSMA/CD) systems, 459–460, 487
- CAs (Certification Authorities)
 - about, 211–213
 - digital certificates using, 83
 - list of, 232
 - on CSP to generate private key and public key pair, 217
- CASE (Computer-Aided Software Engineering)
 - tools, 590, 626
- catalog for backup procedure, 683
- category criteria, definitions in implementing security program, 41
- CBC (Cipher Block Chaining), as mode of symmetric key block ciphers, 181–182
- CBC-MAC (Cipher Block Chaining Message Authentication Code), 187–188
- CC (Common Criteria for Information Technology Security Evaluation), 345, 348–349
- CCITT (International Telegraph and Telephone Consultative Committee), 98
- CCTV (Closed-Circuit TV) system, 267–270
- CDDI (Copper Distributed Data Interface) networks, 447, 454, 487, 489
- CDMA (Code Division Multiple Access), 504
- CEI (Computer Ethics Institute), Ten Commandments of Computer Ethics, 19
- cell phones, attacks on, 513
- cells (data fields), 605
- cellular networking, 504–505
- cellular networks
 - 4G LTE (Long Term Evolution) vs. 4G cellular, 119
 - using Diameter on, 118–119
- centralized access control, 115–119
- Central Processing Unit (CPU)
 - about, 307–308, 307–309
 - Moore's Law and power of, 147–148
 - used in cards for authenticating users, 82
- CER (Crossover Error Rate), in biometric system, 86
- CERT (Computer Emergency Response Team), 386, 564
- Certificate Hold, certificate revocation reason in PKI, 221
- certificate repository, in PKI applications, 220
- Certificate Revocation List (CRL), 220, 221
- certificate vitae (CV), verifying, 46
- Certification and Accreditation (C&A) frameworks, 344–349
- certification and accreditation, of policy documents, 19–20
- Certification Authorities (CAs)
 - about, 211–213
 - digital certificates using, 83
 - list of, 232
 - on CSP to generate private key and public key pair, 217
- Certified Information Systems Auditor (CISA), 335
- Certified Information Systems Manager (CISM), 335
- Certified in Risk and Information Systems Control (CRISC), 335
- Certified in the Governance of Enterprise IT (CGEIT), 335
- CFB (Cipher Feedback mode), as mode of symmetric key block ciphers, 183
- CGEIT (Certified in the Governance of Enterprise IT), 335
- chain of custody, 398
- Challenge Handshake Authentication Protocol (CHAP)
 - about, 484
 - hash value (message digests) and, 91–93
 - Kerberos and, 96
 - MAC types of authentication and, 186
- challenge response, in authentication, 83
- change control (change management)
 - in software development life cycle, 583–584
 - procedures for policy documents, 21
- CHAP (Challenge Handshake Authentication Protocol)
 - about, 484
 - hash value (message digests) and, 91–93
 - Kerberos and, 96
 - MAC types of authentication and, 186
- checklist testing, 567
- checkpoints
 - inserting into data stream, 423
 - in transaction processing, 615

- child pornography, as type of cybercrime, 368
- Chinese wall model, computer system, 343
- chosen plaintext attack, 237
- CIA (Confidentiality, Integrity, Availability) triad
 - about, 5–7
 - losses and, 8
- CIB (Candidate Information Bulletin), 383
- CIDR (Classless Inter-Domain Routing), 435, 477
- CIFS (Common Internet File System), 680
- Cipher-based Message Authentication Code, (CMAC), 189
- Cipher Block Chaining (CBC), as mode of symmetric key
 - block ciphers, 181–182
- Cipher Block Chaining Message Authentication Code, (CBC-MAC), 187–188
- Cipher Feedback mode (CFB), as mode of symmetric key
 - block ciphers, 183
- cipher locks, in target hardening method of designing
 - physical security, 259
- ciphers. *See also* algorithms
 - about, 142–143
 - arbitrary substitution, 151
 - asymmetric key, 148
 - Atbash, 149
 - Caesar or Shift, 150–151
 - concealment, 158
 - Enigma machine, 154–155
 - Feistel Network and, 159
 - group, 151
 - monoalphabetic, 149
 - polyalphabetic, 149
 - running key, 158
 - Scytale, 150
 - substitution, hieroglyphics as, 149
 - symmetric key, 148
 - symmetric key block. *See* symmetric key block
 - ciphers
 - transposition, 145
 - triple DES (3DES), 157, 177–178
 - Vernam, 154
 - Vigenere, 152–153
- ciphertext
 - as attack vector, 165
 - attacks
 - chosen, 237
 - ciphertext-only, 236
 - decrypting, 142
 - encryption converts plaintext message into, 142
 - making strong, 169
 - pattern detection attack on, 145
 - showing patterns of nature of the key, 180
 - using in XOR function, 173
- CIR (Committed Information Rate), 490
- circuit-switched vs. packet-switched networks, 455–456
- CIRT (Computer Incident Response Team), 386
- CISA (Certified Information Systems Auditor), 335
- CISC (Complex Instruction Set Code), 308–309
- CISC (Complex Instruction Set Computing), 595
- CISM (Certified Information Systems Manager), 335
- civil law (tort law), 371, 372
- claim of identity, authentication providing, 140
- Clark-Wilson (CW) model, computer system, 342–343
- classes of fires, 283–284
- classful networks, 476–477
- classification categories, definitions in implementing
 - security program, 40–41
- classifications, in MAC, 105–107
- classifying data, in implementing security program
 - about, 38–39
 - assigning roles and responsibilities, 39–40
- Classless Inter-Domain Routing (CIDR), 435, 477
- cleanroom model, software project-planning models, 590
- clearance label, MAC model, 105, 323
- cleartext, Telnet authentication in, 223
- client and server application architecture, 601–602
- client/endpoint systems, 464–465
- client vs. server, 428
- clipping level thresholds, detection mechanisms, 126
- clock synchronization
 - of KDCs, 98
 - of the devices and systems being monitored, 390
- clone disks, for forensic investigations
 - about, 401
 - analyzing content on, 402–405
 - preparing, 401–402
- Closed-Circuit TV (CCTV) system, 267–270
- cloud computing, 330
- cloud storage services, 679
- cluster IP addresses, 681–682
- clusters, hard disk, 403–404
- CMAC (Cipher-based Message Authentication Code.), 189
- CMMI (Capability Maturity Model Integration), five stages of, 587–588
- CMU (Carnegie Mellon University), 386

CO2 (carbon dioxide) gas, for fire suppression

- CO2 (carbon dioxide) gas, for fire suppression, 287
 - coaxial cable
 - about, 445–446
 - in designing physical security, 263
 - COBIT (Control Objectives for Information and Related Technology), 17, 335
 - Codd, Edgar, 608
 - Code Division Multiple Access (CDMA), 504
 - Code of Ethics, ISC2, 386
 - Code of Federal Regulations, 373
 - code review, in software development, 581
 - Code, US, 373
 - codified law system, 371
 - cognitive passwords, 78
 - cold boot attack, 313
 - cold sites, leased, 545–546
 - collection disk, 404
 - collection of evidence, in evidence life cycle, 397
 - collision occurrences, 459
 - collisions, hashing algorithm resistance to, 167
 - collocation
 - of data, 558
 - of processes, 544
 - of systems, 683
 - collusion, 72, 73, 659
 - combi-card, to authenticate users, 82
 - COM (Component Object Model), 600
 - comma-separated value (.csv) files, 610
 - Committed Information Rate (CIR), 490
 - Committee of Sponsoring Organizations of the Treadway Commission (COSO), 17, 335
 - Common Access Card (CAC) smart card, private key encrypted on, 217
 - Common Criteria for Information Technology Security Evaluation (CC), 345, 348–349
 - Common Internet File System (CIFS), 680
 - common law system, 372
 - Common Vulnerabilities and Exposures (CVEs), 582, 662
 - compartmented security mode, MAC, 108–109
 - compartmented security mode, MAC model, 325
 - compensating controls, 69
 - countermeasures and, 10
 - implementing, 104
 - complementary controls, 10
 - Complex Instruction Set Code (CISC), 308–309
 - Complex Instruction Set Computing (CISC), 595
 - Component Object Model (COM), 600
 - compromise (exploit)
 - losses and, 8
 - quantifying impact of, 9–10
 - Computer-Aided Software Engineering (CASE)
 - tools, 590, 626
 - computer crimes
 - about, 366
 - as percentage of total crime, 369
 - criminal acts involving, 367–368
 - global laws
 - codified law system, 371
 - common law system, 372
 - customary law system, 373
 - governance of third parties, 382–383
 - hybrid law systems, 373
 - laws vs. regulations, 373–374
 - litigation management, 381
 - protecting intellectual property, 374–375
 - protecting privacy, 376–382
 - religious law system, 373
 - software licensing management, 383–384
 - investigating
 - about, 384–385
 - evidence gathering, 396–399
 - forensic investigations, 399–405
 - incident response, 386–395
 - notifying law enforcement, 385–386
 - motivations for, 369
 - prosecution of, 370
 - types of people committing, 369
 - ways computers are involved in, 368
- Computer Emergency Response Team (CERT), 386, 564
- Computer Ethics Institute (CEI), Ten Commandments of Computer Ethics, 19
- Computer Incident Response Team (CIRT), 386
- Computer Security Emergency Response Team (CSERT), 386
- Computer Security Incident Response Team (CSIRT), 386–388
- Computer Security Resource Center (CSRC) SP 800 series, 336–337
- computer systems
 - application architecture, 326–332
 - as information asset, 307–314
 - C&A frameworks, 344–349
 - chart of major components of, 312
 - cold boot attack on, 312

- frameworks for security
 - about, 332–333
 - COBIT, 335
 - COSO, 335
 - GAISP, 336
 - ISO 27000 series, 333–334
 - ITL, 336
 - NIST SP 800 series, 336–337
 - Zachman Framework, 334
- hardware in, 307–314
- legal and regulatory compliance, 349–352
- operating systems
 - about, 314–316
 - buffer overflow attack, 320–321
 - MAC model within, 323–326
 - memory manager, 321–323
 - multiprogramming feature, 316
 - multitasking feature, 317–318
 - multithreading concept, 317
 - processes, 318–320
 - security models, 337–343
- concealment cipher, 158
- concurrency control, 614–615
- CONFIDENTIAL clearance label, 612, 614
- confidentiality
 - agreements. *See* Nondisclosure Agreements (NDAs)
 - breaches of, 7
 - definition of, 5
 - encryption providing, 140
 - in cryptosystem, 143
 - losses and, 8
 - symmetric keys and, 170–171
- configuration management, in software development
 - life cycle, 585
- Congestion Windows Reduced (CWR), TCP segment, 426
- Congress, US, 373
- consistency
 - in transactions, 617
 - of information, 6
 - testing, 567
- constrained (database) view, 612
- constrained interface, access control, 119
- contactless cards, to authenticate users, 82
- contact-oriented cards, to authenticate users, 82
- contact switches, magnetic, in IDS, 274
- containment, in incident response system, 394
- contentionless media access methods, 459
- contention-oriented media access systems, 459
- content journaling, recovering encrypted content using, 163
- Continuance of Operations Plans (COOP), 525
- continuous binary data stream, 422, 423, 424
- control gap, in ALE calculation, 30
- Control Objectives for Information and Related Technology (COBIT), 17, 335
- controls. *See also* access controls
 - administrative access
 - about, 11, 66
 - functional security objectives and, 70
 - MAC and, 108
 - assessment, 68
 - compensating, 10, 69
 - complementary, 10
 - corrective, 68–69
 - countermeasures and, 10–12, 66
 - delay, 67
 - detective, 67–68
 - deterrent, 67
 - directive, 69
 - implementing and maintaining, 12
 - implementing security program and defining required protective, 41–43
 - physical access
 - about, 11, 66
 - functional security objectives and, 70
 - MAC and, 108
 - preventive, 67
 - security, 5–7
 - security through obscurity, 12
 - technical access
 - about, 11, 66
 - DAC as, 109–112
 - functional security objectives and, 70
 - MAC and, 105, 108
 - RBAC, 323
 - role-based access control as, 113
 - rule-based access control, 114–115
- control unit, CPU, 307
- convergence of multiple and redundant data sets, 6
- cookies, 102, 328, 602, 637
- COOP (Continuance of Operations Plans), 525
- cooperative multitasking feature, operating systems, 317
- Copper Distributed Data Interface (CDDI) networks, 447, 454, 487, 489

copyrights

- copyrights, 374–375
- corporate policy framework of governance
 - about, 3–4
 - primary components of, 367
 - types of documents establishing, 15
- corrective controls, 68–69
- COSO (Committee of Sponsoring Organizations of the Treadway Commission), 17, 335
- cost justification for countermeasure
 - managing risk using, 32–33
 - process of performing, 31–32
- cost-justified preventive controls, identifying, 541
- Council of Europe Convention on Cybercrime, 384
- countdown timers, in fire suppression, 287
- countermeasures
 - about, 31
 - controls and, 10–12
 - definition of, 10
 - for reducing or eliminating vulnerabilities, likelihood, and impact, 22–23
 - identifying cost-effective, 31–32
 - implementing and maintaining, 12
 - vs. access controls, 65
- Counter mode (CTR), as mode of symmetric key block ciphers, 183–184
- cover tracks phase, in targeted attacks, 700
- covert storage channel, 627
- covert timing channel, 627
- covert (unintended) communications channels, 627
- CPTED (Crime Prevention through Environmental Design), 252–256
- CPU (Central Processing Unit)
 - about, 307–309
 - Moore’s Law and power of, 147–148
 - used in cards for authenticating users, 82
- CRC (Cyclic Redundancy Check), 437–439
- creation phase, in information life cycle, 37
- credit card data, over Internet, 233, 378–379
- credit card numbers, information theft of, 508
- credit history report, 46
- Crime Prevention through Environmental Design (CPTED), 252–256
- criminal background checks, for prospective employees, 46
- criminal law system, 372
- cripple-ware, 384
- CRISC (Certified in Risk and Information Systems Control), 335
- critical business functions, recovery plans for, 565–566
- CRL (Certificate Revocation List), 220, 221
- Crossover Error Rate (CER), in biometric system, 86
- cross-site scripting (XSS) attacks, 632–634
- cryptanalysis
 - about, 143–146
 - historical review of, 151
- Cryptographic Service Provider (CSP)
 - creation of keys, 160
 - on CA to generate private key and public key pair, 217
- cryptographic services, 143, 156, 170–171, 190–191
- cryptography
 - about, 140–142
 - asymmetric key
 - ciphers, 148
 - distribution of keys, 161
 - ITU-T and, 100
 - length of keys, 160–161
 - management of keys, 159, 191
 - mathematical keys, 202
 - providing nonrepudiation of sender, 140
 - public vs. private key, 191
 - quantities of keys, 162–163
 - SESAME using, 100
 - asymmetric key algorithms
 - about, 170, 190–191
 - list of, 201–207
 - sealing messages using, 195–201
 - signing messages using, 192–195, 198–201
 - attacks on, 184, 236–238
 - authentication providing claim of identity, 140
 - basics of, 142–143
 - components of, 142
 - encryption providing confidentiality, 140
 - end-to-end encryption, 210
 - hashing algorithms/message digests, 165–170
 - historical review of, 148–159
 - keys in, 159–165
 - LAN-based applications, secure channels for, 223–229
 - link encryption, 209–210
 - making strong, 168–169
 - PGP, 221–223
 - PKI
 - as standard for security, 192
 - certificate repository, 220
 - Certification Authorities in, 211

- digital certificates from, 211
- PGP vs., 222
- RAs authenticating user in, 213–214
- trusting certification authority or, 214–215
- X.509 digital certificate, 215–220
- providing integrity of information, 141
- steganography within, 234–235
- symmetric key
 - about, 141
 - Kerberos using, 98
 - nonrepudiation and, 140
 - SESAME using, 100
 - using to delete files securely, 688
- symmetric key algorithms
 - about, 169–170
 - sealing messages using, 185, 189
 - signing using, 185–189
 - weakness in, 189
 - XOR function in, 158, 172–173
- symmetric key cryptography
 - about, 141
 - nonrepudiation of sender and, 140
- Vernam cipher (one-time pad), 154
- web-based applications, secure channels for, 229–234
- cryptoperiod of keys, 164
- cryptosystem
 - brute force attack in cracking, 144
 - five desirable cryptographic functions of, 143, 156, 170–171, 190–191
 - frequency analysis attack on, 145–146
 - length of keys and, 160
 - making strong, 168–169
 - pattern detection attack on, 145
 - performance cost vs. security in, 180
 - PGP as hybrid, 221–222
 - PKI as hybrid, 210–211
 - sealing messages by using asymmetric key algorithms in hybrid, 195–197
 - signing and sending services in
 - symmetric keys and, 171
 - signing messages by using asymmetric key algorithms in hybrid, 192–195
 - social engineering attack on, 145
 - work factor of, 144, 147–148
- cryptosystems
 - Kerckhoffs's principles in, 158
 - XOR function in, 158, 172–173
- cryptovvariable (key), as component of cryptography, 142
- CSERT (Computer Security Emergency Response Team), 386
- CSIRT (Computer Security Incident Response Team), 386–388
- CSMA/CD (Carrier Sense Multiple Access with Collision Detection) systems, 459–460, 487
- CSP (Cryptographic Service Provider)
 - creation of keys, 160
 - on CA to generate private key and public key pair, 217
- CSRC (Computer Security Resource Center) SP 800 series, 336–337
- .csv (comma-separated value) files, 610
- CTR (Counter mode), as mode of symmetric key block ciphers, 183–184
- customary law system, 373
- CV (certificate vitae), verifying, 46
- CVEs (Common Vulnerabilities and Exposures), 582, 662
- CW (Clark-Wilson) model, computer system, 342–343
- CWR (Congestion Windows Reduced), TCP segment, 426
- cyber-attacks, 694
- cybercrime
 - about, 366
 - as percentage of total crime, 369
 - global laws
 - codified law system, 371
 - common law system, 372
 - customary law system, 373
 - governance of third parties, 382–383
 - hybrid law systems, 373
 - laws vs. regulations, 373–374
 - litigation management, 381
 - protecting intellectual property, 374–375
 - protecting privacy, 376–382
 - religious law system, 373
 - software licensing management, 383–384
 - investigating
 - about, 384–385
 - evidence gathering, 396–399
 - forensic investigations, 399–405
 - incident response, 386–395
 - notifying law enforcement, 385–386
 - motivations for, 369
 - prosecution of, 370
 - types of, 367–368

- types of people committing, 369
- ways computers are involved in, 368

cyber-espionage, 694

Cyclic Redundancy Check (CRC), 437–439

Cypherpunks, leaking into public domain RC4, 175

D

DAA (Designated Approving Authority), 20

DAC (Discretionary Access Control)

- about, 109–112
- MAC vs., 323
- operating systems implementing, 109
- permissions provisioning and, 656
- vs. nondiscretionary access control, 113
- Windows Server 2012 as, 108

DAD (Disclosure, Alteration, and Destruction), 7

Daemen, Joan, 157

daemons, 421

daisy-chain outlet strips, 279

Darik's Boot and Nuke (DBAN), 401, 689

darknet, 635

DARPA (Defense Advanced Research Projects Agency), 441–442

data

- backups, 683–687
- increasing value of, 619–624
- retention of, 687–688
- secure deletion of hard disk drive, 688–689

data aggregation, 376, 619

data alteration, as type of cybercrime, 367

data at rest

- about, 7, 141
- in maintenance phase of information life cycle, 38
- in storage phase of information life cycle, 37–38

data backups, 683–687

database (constrained) view, 612

Database Management System (DBMS), 606

database mirroring, 554

database models, 607–610

databases

- exploit, 696–697
- vulnerability, 696

database schemas

- about, 99
- X.500 Directory Services, 99

database shadowing, 554

database systems

- about, 605–607
- accessing databases, 610–612
- artificial intelligence, 620, 621–624
- database models, 605–607
- distributed databases, 618–619
- polyinstantiation, 612–614
- transaction processing, 614–617

database tables

- columns in, 605–606
- data fields in, 605
- rows in, 605
- schema of, 606

data bus, computer system, 311

data center location, as physical control in designing

- physical security, 256

data classification

- assigning enforcement responsibilities, 44–45
- training employees on structure of, 44

data classification program

- assigning roles and responsibilities, 39–40
- defining classification categories, 40–41
- defining protective controls for categories, 41–43

data consumer, 611

data custodian role

- documenting provisioning process, 104
- in classifying data, 39
- in IT, 650

data elements (information assets), inventorying, 43

Data Encryption Algorithm (DEA)

- as symmetric key block cipher, 177
- historical review of, 156

Data Encryption Standard (DES)

- as symmetric key block cipher, 177
- cracking of, 157, 615
- historical review of, 156
- Kerberos using, 94

data fields (cells), 605

datagram, 431

data, increasing value of, 619–624

Data, Information, Knowledge, Wisdom (DIKW) pyramid, 622

data in transit

- about, 7, 141
- in distribution phase of information life cycle, 37
- in maintenance phase of information life cycle, 38

- data in use
 - about, 7, 142
 - in maintenance phase of information life cycle, 38
 - in use phase of information life cycle, 38
- Data level, in knowledge pyramid, 622
- Data Link layer of OSI Model, 435–440, 461–462
- data management
 - classifying data, 39–45, 671
 - fax security, 690–691
 - maintaining systems supporting data, 673–687
 - media management, 672
 - object reuse, 689–690
 - retention of data, 687–688
 - secure deletion of hard disk drive data, 688–689
 - secure destruction policy, 689–690
- data mart, 621
- data mining, 620–621
- data normalization, 620
- data normalization, in application architecture, 326
- data origin, authentication based on symmetric keys, 185
- data owner
 - documenting provisioning process, 104
 - in account creation, 71
 - role in classifying data, 39
- data owner role
 - in IT, 650
- Data Protection Directive (EU), 376–377
- data provider, 611
- data repositories, in application architecture, 326
- data streams
 - continuous binary, 422, 423, 424
 - segment, 424
- data theft, as type of cybercrime, 367
- data warehouse, 620
- DBAN (Darik's Boot and Nuke), 401, 689
- DBMS (Database Management System), 606
- DC (Domain Controller), Active Directory, 95
- DCOM (Distributed Component Object Model), 600
- dd (Linux commandline utility), 402
- DDoS (Distributed Denial of Service) attack, 471, 507–508
- DEA (Data Encryption Algorithm)
 - as symmetric key block cipher, 177
 - historical review of, 156
- decentralized access control, 115
- decryption
 - ciphertext, 142
 - escrow of keys, 163
 - stored data on smart cards, 82
- dedicated security mode, MAC, 108
- dedicated security mode, MAC model, 325
- Deep Crack, 157
- default gateway, as part of IP network, 431–432
- Defense Advanced Research Projects Agency (DARPA), 441–442
- defense in depth
 - about, 10
 - implementing compensating controls in, 104
- delay controls, 67
- delaying intruders, 250, 251
- deliverables of project
 - defining scope of project, 531
 - scheduling, 532–533
- deliveries requirement, in recovery plan, 549
- Delphi Method, 28
- deluge sprinkler systems, 285–286
- Demilitarized Zone (DMZ), 69
- demonstrative evidence, 398
- denial of service (DoS), 367–368, 506–507, 664
- Deny All rule, 467
- Department of Defense Computer Security Center, US, IT security standards and guidelines by, 345
- Department of Homeland Security, US, risk assessment on DNS systems, 474
- depth of field of camera, in designing physical security, 269
- DES (Data Encryption Standard)
 - as symmetric key block cipher, 177
 - cracking of, 157, 615
 - historical review of, 156
 - Kerberos using, 94
- Designated Approving Authority (DAA), 20
- detecting intruders, 250, 251
- detection mechanisms
 - IDS/IPS, 125–126
 - in incident response system, 391
 - malware, 637–638
- detective control, 67–68
- detectors, fire, 282–283
- deterministic media access methods, 459
- deterrent controls, 67, 69
- deterring intruders, 250, 251
- device drivers, downloading, 316

device-related mechanisms, authentication

- device-related mechanisms, authentication
 - about, 81–83
 - drawbacks of, 83–84
 - DHCP (Dynamic Host Configuration Protocol), 474–475
 - DH (Diffie-Hellman) algorithm, 156, 161, 190, 202–205
 - diagnostic port, sensors attaching to layer-2 MAC switches using, 128
 - dial-in connections, authentication accessing from, 90–91
 - Diameter, 116, 118–119
 - dictionary attack, on passwords, 80, 144
 - differential backup, 556–557, 684, 685–686
 - Diffie-Hellman (DH) algorithm, 156, 161, 190, 202–205
 - Diffie, Whitfield, 190
 - digital certificates
 - intended purposes of, 217
 - ITU-T and, 100
 - to authenticate users, 83
 - digital encoding, 452–453
 - Digital Signal Processors (DSPs), 497
 - Digital Signature Algorithm (DSA), 206–207
 - digital signatures
 - about, 192
 - as standard for security, 192
 - MACs vs., 186
 - using asymmetric key algorithms, 193–194
 - Digital Signature Standard (DSS), 206–207
 - Digital Subscriber Line (DSL), 488
 - DIKW (Data, Information, Knowledge, Wisdom) pyramid, 622
 - direct evidence, 398
 - directive controls vs. deterrent, 69
 - direct manager, in account creation, 71
 - directory transversal attacks, 636
 - Direct Sequence Spread Spectrum (DSSS), 496–497
 - disabling or locking down user accounts, 71
 - Disaster Recovery Plan (DRP)
 - about, 4, 527
 - components of plans, 563–569
 - developing plan proposals
 - about, 540
 - alternative leased sites, 545–546
 - backup strategies and storage, 555–559
 - collocation of processes, 544
 - compliance with laws and regulations, 542
 - considering alternative procedures, 542
 - identifying preventive controls, 541
 - reciprocal agreements, 546–547
 - recognizing increased operating costs, 542
 - recovery of data, 551–554
 - recovery of personnel, 559–560
 - recovery of supply systems, 547–548
 - recovery of technologies, 548–550
 - rolling hot sites, 546
 - security standards, 550
 - workspace recovery, 543–544
 - development of, 525–526
 - identifying single points of failure, 655
 - implementing approved plans, 562–563
 - presentation to senior management, 561–562
 - sharing accomplishment, 570
 - stages of planning process
 - about, 529–530
 - defining need, 530
 - defining planning budget and schedule, 532–533
 - defining planning project leader, 530
 - defining planning scope of project, 531
 - defining planning team, 531
 - performing business impact analysis, 533–540
 - timeline for, 529
- disasters, 525–526
- Discover, Offer, Request, Acknowledgement (DORA), 474
- discovery phase of court case, 397
- Discretionary Access Control (DAC)
 - about, 109–112
 - MAC vs., 323
 - operating systems implementing, 109
 - permissions provisioning and, 656
 - vs. nondiscretionary access control, 113
 - Windows Server 2012 as, 108
- discrimination, avoiding lawsuits for, 15
- disk duplex, 676
- disk encryption, on portable devices, 271
- disk mirroring (RAID), 553, 674–677
- disk shadowing, 554
- disposal and end of life, in software development life cycle, 585–586
- disposal phase, in information life cycle, 38
- distance vector dynamic routing protocols, 481
- Distributed Component Object Model (DCOM), 600
- distributed computing, 599–604, 618
- distributed databases, 618–619
- distributed data processing, 618

- Distributed Denial of Service (DDoS) attack, 471, 507–508
 - distributed systems, in SOA, 329
 - distribution phase, in information life cycle, 37
 - distribution restrictions, protective controls for, 42
 - DNS (Domain Name System), 434, 473–474, 480
 - DNS poisoning, information theft by, 510–511
 - DNSSEC (Domain Name System Security Extensions), 474
 - documentary evidence, 398
 - documentation
 - in evidence life cycle, 396
 - of networking environment of enterprise, 549
 - Document Object Model (DOM), cross-site scripting attack, 634
 - Domain Controller (DC), Active Directory, 95
 - domain controller (Microsoft Windows), authentication service running on, 75
 - Domain Name System (DNS), 434, 473–474, 480
 - Domain Name System Security Extensions (DNS-SEC), 474
 - domain (realm), Kerberos, 95
 - DOM (Document Object Model), cross-site scripting attack, 634
 - doors, in target hardening method of designing physical security, 258
 - Doppler effect, IDEs using, 273
 - DORA (Discover, Offer, Request, Acknowledgement), 474
 - DoS (denial of service), 367–368, 506–507, 664
 - double DES (2DES), 177
 - downstream liabilities, 381
 - DRAM (Dynamic Random Access Memory), 310, 312
 - drive-by downloads, 630–631
 - DRP (Disaster Recovery Plan)
 - about, 4, 527
 - developing plan proposals
 - about, 540
 - alternative leased sites, 545–546
 - backup strategies and storage, 555–559
 - collocation of processes, 544
 - considering alternative procedures, 542
 - identifying preventive controls, 541
 - reciprocal agreements, 546–547
 - recognizing increased operating costs, 542
 - recovery of data, 551–554
 - recovery of personnel, 559–560
 - recovery of supply systems, 547–548
 - recovery of technologies, 548–550
 - rolling hot sites, 546
 - security standards, 550
 - workspace recovery, 543–544
 - development of, 525–526
 - identifying single points of failure, 655
 - implementing approved plans, 562–563
 - presentation to senior management, 561–562
 - sharing accomplishment, 570
 - stages of planning process
 - about, 529–530
 - defining need, 530
 - defining planning budget and schedule, 532–533
 - defining planning project leader, 530
 - defining planning team, 531
 - performing business impact analysis, 533–540
 - timeline for, 529
 - drug-screening, prospective employees, 46
 - dry chemicals, for fire suppression, 288
 - dry pipe sprinkler systems, 285
 - dry powder, for fire suppression, 290
 - DSA (Digital Signature Algorithm), 206–207
 - DSL (Digital Subscriber Line), 488
 - DSPs (Digital Signal Processors), 497
 - DSS (Digital Signature Standard), 206–207
 - DSSS (Direct Sequence Spread Spectrum), 496–497
 - dual control, deterring fraud using, 73, 659
 - due care, implementing, 12, 381
 - due diligence, performing, 12, 381
 - dumb terminals (thin clients), 454, 463
 - dumpster diving, as type of cybercrime, 368, 665
 - durability, in transactions, 617
 - dynamic binding, in object-oriented programming, 598
 - Dynamic Host Configuration Protocol (DHCP), 474–475
 - dynamic packet filtering, generation 4 firewall, 469
 - Dynamic Random Access Memory (DRAM), 310, 312
 - dynamic routing protocol, 462, 481–482
 - dynamic separation of duties, deterring fraud using, 73, 659
- E**
- E1 connections, 490
 - E3 connections, 490
 - EAL (Evaluation Assurance Level), as component in evaluation and certification process, 349

EAP (Extensible Authentication Protocol)

- EAP (Extensible Authentication Protocol), 101, 224, 484
- eavesdropping/sniffing, information theft by, 508
- ECB (Electronic Code Book), as mode of symmetric key block ciphers, 180–181
- ECC (Elliptic Curve Cryptography), 205–206
- ECE (Explicit Congestion Notification (ECN) Echo), TCP segment, 426
- Eckert, John, 190
- e-commerce websites, credit card purchases on, 233
- ECSTR (Efficient and Compact Subgroup Trace Representation), 207
- EDGAR (Electronic Data Gathering, Analysis, and Retrieval) database, 694
- EDI (Electronic Data Exchange), 351
- eDirectory, X.500 Directory Services and, 99
- EEOC (Equal Opportunity Employment Commission), on discrimination-related lawsuits, 15
- EEPROM (Electrically Erasable, Programmable, Read-only Memory), 310
- EF (Exposure Factor), 29
- Efficient and Compact Subgroup Trace Representation (ECSTR), 207
- egress and ingress filter firewall rules, 470
- EICAR (European Institute for Computer Antivirus Research) Test Files, 629
- EIGRP (Enhanced Interior Gateway Routing Protocol), 481
- Electrically Erasable, Programmable, Read-only Memory (EEPROM), 310
- Electrical Numerical Integrator And Computer (ENIAC), 190
- electricity considerations
 - in designing physical security, 277–279
 - in developing business continuity plans, 547–548
- Electromagnetic Interference (EMI), 262, 278, 444
- electromechanical cryptographic machine, 154
- Electronic Code Book (ECB), as mode of symmetric key block ciphers, 180–181
- Electronic Data Exchange (EDI), 351
- Electronic Data Gathering, Analysis, and Retrieval (EDGAR) database, 694
- electronic locks, in target hardening method of designing physical security, 259
- electronic vaulting, 558
- ElGamal, 206
- Elliptic Curve Cryptography (ECC), 205–206
- emanations detection, information theft by, 508
- emanations protection, in designing physical security, 261–263
- emanations, reducing, 443–444
- Emanations Security (EMSEC)
 - media types for, 449
 - Tempest classified as, 444
- Emergency Power Off (EPO) button, 289
- emergency services, evaluating proximity of, 253
- emergency system restart, 346
- EMI (Electromagnetic Interference), 262, 278, 444
- employee privacy issues, monitoring environment and, 379–380
- employees
 - drug-screening prospective, 46
 - exit interview of, 47–48
 - lawsuits and workplace crime involving, 45
 - providing security awareness training program to, 44, 49
 - standard of behavior and activity of, 15–16
- Encapsulating Security Payload (ESP), 227
- encapsulation of objects, in object-oriented programming, 598–599
- encoding data into signals, 450–453
- encrypted channel, privacy issues in trans-border information flow over, 380
- encrypting viruses, 629
- encryption
 - converts plaintext message into ciphertext, 142
 - end-to-end, 210
 - link, 209–210
 - of stored data on smart cards, 82
 - providing confidentiality, 140
 - requirements, protective controls for, 42
 - retrieving content after, 163
 - whole disk, 313
- encryption tools
 - BitLocker Drive Encryption, 141
 - TrueCrypt application as, 141
- endpoint/client systems, 464–465
- end-to-end encryption, 210
- End-User License Agreements (EULA), 383
- enforcer role, in classifying data, 39
- Enhanced Interior Gateway Routing Protocol (EIGRP), 481
- ENIAC (Electrical Numerical Integrator And Computer), 190
- Enigma machine, 154
- enrollment process, for biometric system, 85
- enticement, 130, 398
- entity integrity validation, 609

entrapment, 130, 398
 entrench phase, in targeted attacks, 699
 entryway barriers, in designing physical security, 265–266
 environmental design, crime prevention through, 252–256
 ephemeral port numbers, 430
 EPL (Evaluated Products List), as component in evaluation and certification process, 349
 EPO (Emergency Power Off) button, 289
 Equal Opportunity Employment Commission (EEOC), on discrimination-related lawsuits, 15
 Erasable Programmable Read-Only Memory (EPROM), 310, 463
 escalation process, 391
 escrow of decryption keys, 163
 escrow, software, 593–594
 ESP (Encapsulating Security Payload), 227
 eSTREAM, 175
 Ethernet networks
 about, 487
 hazardous, 7
 running over UTP cables, 447, 454
 STP cabling, 447
 ethical standards, policy documents conveying, 18–19
 “Ethics and the Internet” (IAB), 19
 EU (European Union), Data Protection Directive, 376–377
 EULA (End-User License Agreements), 383
 European Institute for Computer Antivirus Research (EICAR) Test Files, 629
 European Union (EU), Data Protection Directive, 376–377
 evacuation routes, fire, 291
 Evaluated Products List (EPL), as component in evaluation and certification process, 349
 Evaluation Assurance Level (EAL), as component in evaluation and certification process, 349
 EV-DO (Evolution-Data Optimized (or Data Only)), 504
 evidence gathering, in investigating computer crimes, 396–399
 evidence life cycle, 396–397
 Evolution-Data Optimized (or Data Only) (EV-DO), 504
 examination and analysis, in evidence life cycle, 397
 Exclusive Or (XOR) function, 158, 172–173, 175–176, 178
 executable files, 596, 604
 executable processes, quarantining malicious or suspicious, 124–125

executive succession, 560
 executive summary, in penetration testing report, 669
 exit interview, 47–48
 exit signs, 266
 expert systems, 623
 Explicit Congestion Notification (ECN (ECE) Echo), TCP segment, 426
 explicit permissions, 114
 exploit code, 629
 exploit (compromise)
 losses and, 8
 quantifying impact of, 9–10
 exploit phase, in targeted attacks, 696–697
 Exposure Factor (EF), 29
 exposure, identifying asset, 10
 expression of ideas, protection of, 374
 eXtended TACACS (XTACACS), 117
 Extensible Authentication Protocol (EAP), 101, 224, 484
 Extensible Markup Language (XML), 102, 603, 612
 extension cords, 279
 external threats, 9
 extranet, 458, 482

F

fail safe and fail secure security structure, 266
 failure recovery considerations in designing physical security, 275–279
 False acceptance (Type II error), in biometric system, 85
 False Rejection Rate (FRR), in biometric system, 86
 False rejection (Type I error), in biometric system, 85
 Faraday cage, as emanations protection, 263, 508
 FAT (File Allocation Table), 402, 688
 fault, electrical, 277
 fault tolerance
 improving, 673–674
 in recovery plan, 549–550, 655
 fax line encryption device, 690–691
 fax security, 690–691
 FCC (Federal Communications Commission)
 administrative law and, 372
 regulation of wireless networks, 449
 FCS (Frame Check Sequence), 437
 FDA (Food and Drug Administration), 372
 FDD (Frequency-Division Duplexing), 504
 FDDI (Fiber Distributed Data Interface) networks, 454, 487, 489

FDMA (Frequency-Division Multiple Access)

- FDMA (Frequency-Division Multiple Access), 504
- FDM (Frequency-Division Multiplexing), 457
- FE-13, for fire suppression, 287
- feature enhancements, in software development, 582
- Federal Communications Commission (FCC)
 - administrative law and, 372
 - regulation of wireless networks, 449
- Federal Emergency Management Agency (FEMA), 29
- Federal Information Processing Standard (FIPS)
 - AES standard, 157, 178
 - CBC-MAC approval of, 188
 - DEA and DES standard, 156
 - DSS adopted by, 206
 - ECC approval for protecting secret or classified information, 206
- Federal Information Security Management Act (FISMA), 17, 377
- Federal Trade Commission (FTC), 372
- federated identity management, 76
- Feistel Network, 159
- felonies, 372
- FEMA (Federal Emergency Management Agency), 29
- fences, in target hardening method of designing physical security, 259–261
- fetch function, CPU control unit, 307, 308–309
- FHSS (Frequency-Hopping Spread Spectrum), 486, 495–496
- Fiber Distributed Data Interface (FDDI) networks, 454, 487, 489
- fiber optic cables, 263, 448, 508
- field of view of camera, in designing physical security, 269
- Fighting Computer Crime (Parker), 248
- File Allocation Table (FAT), 402–404, 688
- file named hosts, 473
- File Transfer Protocol (FTP), 233, 479
- finding matching record, in biometric system, 87–89
- FIN (Final), TCP segment, 426, 428
- fingerprint, authentication using, 84
- fingerprinting of nodes, 695
- finger scan, authentication using, 84
- FIPS (Federal Information Processing Standard)
 - AES standard, 157, 178
 - CBC-MAC approval of, 188
 - DEA and DES standard, 156
 - DSS adopted by, 206
 - ECC approval for protecting secret or classified information, 206
- fire code, dictating number and location of doors, 254
- fire, physical security and
 - about, 281
 - fire extinguishers, 288–291
 - four legs of fire, 281–282
 - plan and drill, 291–292
 - sprinkler systems, 284–286
 - suppression agents, 286–290
 - types of detectors, 282–284
- firewall rule, as rule-based access control, 114–115
- firewalls
 - as network devices, 467–471
 - demilitarized zone, 69
 - five generations of, 468–469
 - multihomed, 469–470
 - network sensors in and outside external, 128
 - screened subnet, 69
 - using, 469–471
- firmware, 306
- FISMA (Federal Information Security Management Act), 17, 377
- flash memory (USB) drives, 310, 689
- FM-200, for fire suppression, 287
- focal length of camera lens, in designing physical security, 269
- Food and Drug Administration (FDA), 372
- foot candles, 267
- footprinting target network and nodes, 695
- Foremost, recovering remnants using, 404
- forensic field kit, 399–400
- forensic investigations, in computer crimes
 - about, 399–400
 - analysis of, 400–401
 - clone disks for
 - analyzing content on, 402–405
 - preparing, 401–402
- forward chaining, 623
- 4G LTE (Long Term Evolution) vs. 4G cellular, 119
- four methods of managing risk, 32–33
- FQDNs (Fully Qualified Domain Names), 434, 473
- Fraggle attack, DoS, 506
- Fragmentation (Frag) attack, DoS, 506
- Frame Check Sequence (FCS), 437
- frame relay, 380, 490
- frames, 436
- framework of governance, corporate policy
 - about, 3–4
 - primary components of, 367
 - types of documents establishing, 15

frameworks and policies, 14–15

frameworks for security

- about, 332–333
- COBIT, 335
- COSO, 335
- GAISP, 336
- ISO 27000 series, 333–334
- ITL, 336
- NIST SP 800 series, 336–337
- Zachman Framework, 334

fraud

- about, 342
- authority leading to, 104
- CW model targeting control of, 342
- managing, 72–73
- protection, 657–661

Fraud as a Service (FaaS), 51

FreeBSD operating system, MAC implementation in, 107

free space, clone disk, 403–404

frequency analysis attack, 145–146

Frequency-Division Duplexing (FDD), 504

Frequency-Division Multiple Access (FDMA), 504

Frequency-Division Multiplexing (FDM), 457

Frequency-Hopping Spread Spectrum (FHSS), 486, 495–496

Frequency Modulation (FM), 451

FTC (Federal Trade Commission), 372

FTP (File Transfer Protocol), 233, 479

FTP over a Secure Sockets Layer tunnel (FTPS), 479

FTP over SSL (FTPS), 233

FTP traffic, establishing baseline normal level of, 126

full backup, 555, 557, 684

full disclosure testing, 667

full-duplex, 423

full interruption testing, 567

full wall vs. partition, in target hardening method of designing physical security, 257

Fully Qualified Domain Names (FQDNs), 434, 473

functional design, in software development life cycle, 580

functionality, in computer system security, 344

functional security objectives, 70

fuzzy data and fuzzy logic, 623

G

G8 (Group of 8), 384

GAISP (Generally Accepted Information Security Principles), 336

GAN (Global Area Network), 458

garbage collection, 319, 626

gases, for fire suppression, 286

gateway function, 421, 463

Generally Accepted Information Security Principles (GAISP), 336

generations of firewalls, 468–469

generations of programming languages, 596–597

generators, using in designing physical security, 276

Generic Routing Encryption (GRE), 224

Germany, using Enigma machine during World War II, 154

GLBA (Gramm Leach Bliley Act) of 1999, 17, 351, 372, 377

Global Area Network (GAN), 458

global laws, computer crime

- codified law system, 371
- common law system, 372
- customary law system, 373
- governance of third parties, 382–383
- hybrid law systems, 373
- laws vs. regulations, 373–374
- litigation management, 381
- protecting intellectual property, 374–375
- protecting privacy, 376–382
- religious law system, 373
- software licensing management, 383–384

Global System for Mobile Communications (GSM), 504

goals and objectives, presented in plans, 5

going after the low-hanging fruit attacks, 693

gold standard system, comparing system configuration to, 661

governance

- documents building framework for, 15–16
- of third-party service providers, 382–383
- sources for developing documents, 16–18

governance, security

- due care, 12
- due diligence, 12
- foundation for, 4

GPO (Group Policy Object), 122

Graham-Denning model, computer system, 343

Gramm Leach Bliley Act (GLBA) of 1999

Gramm Leach Bliley Act (GLBA) of 1999, 17, 351, 372, 377
grazing attacks, 693
GRE (Generic Routing Encryption), 224
grid computing, 331–332
group ciphers, Shift (Caesar) cipher as, 151
Group of 8 (G8), 384
Group Policy Object (GPO), 122
GSM (Global System for Mobile Communications), 504
guard dogs, as part of designing physical security, 265
guideline documents, 16

H

hacktivism, 366, 368
HA (High Availability), 95
half-duplex, 423
halon gases and alternatives, for fire suppression, 286–287
hand geometry, authentication using, 84
handling procedures and technologies, protective controls for, 42
hand topology, authentication using, 84
hard disk
 blocks (clusters) on, 403–404
 elimination of remnants, 401, 404
 file allocation table, 402
 free and slack space, 403–404
 hidden content on, 405
 reusing, 689–690
 secure deletion of data from, 688–689
 tracks and sectors on, 403
hardened systems/bastion host systems, 465–467
hardware, computer, 307–314
hardware guard, access control, 119
Hardware Security Modules (HSMs), storing and retrieving escrowed keys, 163
hardware tap, 128
Hashed Message Authentication Code (HMAC), 186–187
hashing algorithms
 about, 165–168
 attacks on, 167–168
 historical review of, 155–156
 in preparing clone disk for forensic investigations, 402
 output lengths of, 166–167
 storing passwords through, 79
hash values (message digests)
 about, 144, 165–168
 attacks on, 167–168
 CHAP open-standard and, 91–92
 hashing algorithms generating, 79
 output lengths of, 166–167
 virtual password and, 78
hazardous network connections, 7, 141
HDD/Erase, 401, 689
HDLC (High-Level Data Link Control), 463
Healthcare Insurance Portability and Accountability Act (HIPAA) of 1996, 17, 351, 372, 377
hearsay rule, 399
heartbeat, server application, 681–682
Heating, Ventilation, and Air Conditioning (HVAC) systems, 274–275, 547
heavy timber construction method of facility, as physical control in designing physical security, 255
Hebern rotor-based system, 154
Helix distribution disk, 402, 404
Hellman, Martin, 156, 190, 207
heuristic-based malware detection, 638
hidden content, on clone disk, 405
hidden (covert) communications channels, 627
hidden partitions, 700
HIDs (Human Interface Devices), 125, 128, 312
hierarchical database model, 607–608
Hierarchical Storage Management (HSM), 679, 680, 687
hieroglyphics, 149
High Availability (HA), 95
high cohesiveness of objects, 599
high-speed fiber channels, 679
hijacking webpages, 635
HIPAA (Healthcare Insurance Portability and Accountability Act) of 1996, 17, 351, 372, 377
hiring practices, implementing, 45–47, 658
HMAC (Hashed Message Authentication Code), 186–187
holddown timers, 482
hold harmless, in pen testing agreement, 668
Homeland Security, US Department of, risk assessment on DNS systems, 474
honeynets, 129–130
honeypots, 129–130
host-based firewall (personal firewall), 467
host-based IDS (HIDS or HIPS), 125, 128

host-based IPS detection systems, 126
 host executable, 629
 hostnames, 473
 hosts, file named, 473
 hosts files, 422
 host-to-host VPN connectivity, 224–225
 host-to-subnet VPN connectivity, 224–225
 hot sites
 leased, 545
 rolling, 546
 HR (Human Resources)
 documenting provisioning process, 104
 in account creation, 71
 provisioning request initiated, 656
 HSM (Hierarchical Storage Management), 679, 680, 687
 HSMs (Hardware Security Modules), storing and retriev-
 ing escrowed keys, 163
 HTTP-GET request, 102
 HTTP (Hypertext Transfer Protocol), 102, 480
 HTTP over SSL (HTTPS), 230–232, 480
 Human Interface Devices (HIDs), 125, 128, 312
 human-made threats
 about, 9, 247, 539
 in performing risk assessment, 29
 Human Resources (HR)
 documenting provisioning process, 104
 in account creation, 71
 provisioning request initiated, 656
 humidity and temperature considerations, in designing
 physical security, 274–275
 HVAC (Heating, Ventilation, and Air Conditioning) sys-
 tems, 274–275, 547
 hybrid access control, 115
 hybrid attack, on passwords, 80, 144
 hybrid card, to authenticate users, 82
 hybrid cryptosystems
 about, 239
 PGP as, 221–222
 PKI as, 210–211
 sealing messages by using asymmetric key algo-
 rithms in, 195–197
 signing messages by using asymmetric key algo-
 rithms in, 192–195
 hybrid law systems, 373
 Hypertext Transfer Protocol (HTTP), 102, 480
 hypervisor, 329

I
 IaaS (Infrastructure as a Service), 331, 679
 IAB (Internet Architecture Board)
 canons of code of ethics, 19
 “Ethics and the Internet”, 19
 IANA (Internet Assigned Numbers Authority), 428, 430
 ICANN (Internet Corporation for Assigned Names and
 Numbers), 458
 IC (Integrated Circuit) chips, 307
 ICMP (Internet Control Message Protocol), 479
 ID badges, photo, to authenticate users, 81
 IDEA (International Data Encryption Algorithm)
 as symmetric key block cipher, 179
 rounds of cryptographic processing, 175
 IDEAL model, software development, 588
 IDEA NXT, 179
 identification of evidence, in evidence life cycle, 396
 identification (one-to-many) method, in biometric
 system, 87
 identification process, 74–75
 identity-based access control, DAC as, 110
 Identity Management (IdM), 76
 IDF (Israel Defense Force), launching Stuxnet, 368
 IdM (Identity Management), 76
 IDS (Intrusion Detection Systems)
 about, 124
 acoustic sensors in, 273
 architecture, 127
 components of, 126
 designing physical security using, 272–274
 detection mechanisms used by, 125–126
 entrenching and, 699
 for incident response, 391
 magnetic contact switches in, 274
 mechanisms to defeat sensors, 127
 monitoring and auditing in, 52
 monitoring output, 390
 network-based and host-based, 125
 photoelectric sensors in, 273
 pressure mats in, 274
 proximity detectors in, 273
 response system, 125
 signatures, 591
 vs. intrusion prevention systems and, 125
 IEC (International Electro-technical Commission) 27000
 series, 17

- IEEE (Institute for Electrical and Electronics Engineers)
 - 802 specifications, 440
 - Data Link layer and, 435–436
- IETF (Internet Engineering Task Force)
 - code of ethics, 19
 - defining HTTPS, 230
 - defining S-HTTP, 232
 - defining SSL, 230
 - introducing Internet Protocol version 6, 433
 - IPsec defined in, 224
 - publishing L2TP, 228
 - publishing OpenPGP, 222
- if-then reasoning, 623
- IGMP (Internet Group Management Protocol), 476, 478
- IGRP (Interior Gateway Routing Protocol), 481
- IKE (Internet Key Exchange), 225–226
- IMAP4, 480
- IM (Instant Messaging), information theft using, 511
- impact
 - countermeasures for reducing or eliminating, 22–23, 31
 - quantifying compromise, 9–10
- implementing security program
 - about, 34
 - assigning enforcement responsibilities, 44–45
 - assigning value to assets, 43
 - classifying data in, 38–41
 - components in, 34
 - defining category criteria, 41
 - defining classification categories, 40–41
 - defining required protective controls, 41–43
 - elemental phases in information life cycle, 37–38
 - implementing hiring practices, 45–47
 - implementing termination practices, 47–48
 - in risk assessment, 34
 - inventorying information assets, 43
 - managing third-party service providers, 50–51
 - monitoring and auditing in, 51–54
 - providing security awareness training program, 44, 49
 - reappraising and adjusting classification of information assets, 44
 - understanding organization chart, 36–37
- incident response, 670–671
- incident response system
 - analysis in, 394
 - containment in, 394
 - creating CSIRT plan, 387–389
 - investigation in, 393–394
 - monitoring in, 389–391
 - notification in, 392
 - prevention in, 395
 - recovery in, 395
 - reporting in, 395
 - tracking in, 394
 - training CSIRT, 387
 - triage in, 393
- incremental backup, 556, 684–687
- inference engine, 623
- information assets
 - assigning value to, 43
 - computer systems as, 307–314
 - definition of, 5
 - inventorying, 43
 - reappraising and adjusting classification of, 44
 - vulnerabilities of, 8
 - vulnerabilities within, 9
- information flow model, computer systems, 339
- Information level, in knowledge pyramid, 622
- information life cycle, elemental phases in, 37–38
- information security
 - protecting privacy, 376–382
 - risk management and, 2–3
- Information Security Management Systems (ISMS), 333–334
- Information System Risk Management (ISRM), 334
- information systems
 - auditing for protection of, 379–382
 - vulnerabilities within, 9
- Information Systems Audit and Control Association (ISACA), 335
- Information Systems Security Association (ISSA), 336
- Information Technology Infrastructure Library (ITIL), 336
- Information Technology (IT). *See* headings in index starting with IT
- Information Technology Security Evaluation Criteria (ITSEC), 347–348
- information theft, 508–511
- Infrared Data Association (IrDA), 485
- infrared fire detectors, 283
- infrared night vision monitoring camera, 268
- Infrastructure as a Service (IaaS), 50, 331, 679
- infrastructure mode, wireless network, 495
- ingress and egress filter firewall rules, 470

- inheritance feature of objects, 598
- in-house inventory, 549
- Initialization Vectors (IVs), 165, 172
- Initial Program Load (IPL), 463
- in-person authentication in PKI system, 214
- inrush, electrical, 277
- inspection, fire extinguisher, 290–291
- inspections and walkthroughs, performing periodic, 279–280
- instantiating class into object, 598
- Instant Messaging (IM), information theft using, 511
- Institute for Electrical and Electronics Engineers (IEEE)
 - 802 specifications, 440
 - Data Link layer and, 435–436
- instruction set, processor, 595
- intangible assets, in performing risk assessment, 25
- Integrated Circuit (IC) chips, 307
- Integrated Services Digital Network (ISDN), 490
- integration testing, in software development, 567, 580, 590
- integrity
 - breaches of, 7
 - definition of, 6
 - in cryptosystem, 143
 - losses and, 8
- integrity of information, components of, 141
- integrity protection
 - about, 6
 - cryptography providing, 141
- integrity validation
 - checks, 609
 - malware detection and, 638
 - provided by digital signatures, 192
 - symmetric keys and, 171
- integrity verification
 - about, 6
 - cryptography providing, 141
 - in fraud protection, 658
- Intellectual Property (IP), 367, 368, 374–375
- intended communications channel (overt), 627
- Interior Gateway Routing Protocol (IGRP), 481
- internal auditing, 122
- internal threats, 9
- International Data Encryption Algorithm (IDEA)
 - as symmetric key block cipher, 179
 - rounds of cryptographic processing, 175
- International Electro-technical Commission (IEC) 27000 series, 17
- International Organization for Standardization (ISO)
 - 15408 series as CC, 348
 - 27000 series, 17, 333–334
 - about, 417
 - CC broken into parts by, 349
 - on OSI model, 418
- International Organization on Computer Evidence (IOCE), 384
- International Telecommunication Union Telecommunication Standardization Sector (ITU-T), 98–100, 215, 490
- International Telegraph and Telephone Consultative Committee (CCITT), 98
- Internet
 - bots, 635
 - credit card data over, 233, 378–379
 - dark address space, 477
 - dark space, 635
 - data in transit protection over, 141
 - network classification, 458
 - privacy issues in trans-border information flow over, 380
 - securing channels for web-based applications, 229–234
- Internet Architecture Board (IAB)
 - canons of code of ethics, 19
 - “Ethics and the Internet”, 19
- Internet Assigned Numbers Authority (IANA), 428, 430
- Internet Control Message Protocol (ICMP), 479
- Internet Corporation for Assigned Names and Numbers (ICANN), 458
- Internet Engineering Task Force (IETF)
 - code of ethics, 19
 - defining HTTPS, 230
 - defining SSL, 230
 - “Ethics and the Internet”, 19
 - introducing Internet Protocol version 6, 433
 - IPsec defined in, 224
 - publishing L2TP, 228
 - publishing OpenPGP, 222
- Internet Group Management Protocol (IGMP), 476, 478
- Internet Key Exchange (IKE), 225–226
- Internet Protocol Security (IPsec)
 - about, 7, 483
 - as VPN technology
 - AH, 226–227
 - IKE, 225–226
 - in Transport mode, 227
 - in Tunnel mode, 228

Internet Protocol version 4 (IPv4)

- data in transit protection using, 141
- in IPv6, 478
- Internet Protocol version 4 (IPv4), 224, 432–433, 471, 475–477
- Internet Protocol version 6 (IPv6), 224, 433, 471, 477–478
- Internet Research Task Force (IRTF), code of ethics, 19
- Internet Security Association protocol, 226
- Internet Service Providers (ISPs), 488
- Interpol, 385
- interpreter, 596
- intranet, 457
- Intrusion Detection Systems (IDS)
 - about, 124
 - acoustic sensors in, 273
 - architecture, 127
 - components of, 126
 - designing physical security using, 272–274
 - detection mechanisms used by, 125–126
 - entrenching and, 699
 - for incident response, 391
 - magnetic contact switches in, 274
 - mechanisms to defeat sensors, 127
 - monitoring and auditing in, 52
 - monitoring output, 390
 - network-based and host-based, 125
 - photoelectric sensors in, 273
 - pressure mats in, 274
 - proximity detectors in, 273
 - response system, 125
 - signatures, 591
 - vs. intrusion prevention systems and, 125
- Intrusion Prevention Systems (IPS)
 - about, 124
 - architecture, 127
 - attackers using, 129
 - detection mechanisms used by, 125–126
 - monitoring and auditing in, 52
 - monitoring output, 390
 - types of response to detect attacks, 124
 - vs. intrusion detection systems, 125
- inventory assets, in performing risk assessment, 24–25
- investigating computer crimes
 - about, 384–385
 - incident response, 386–395
 - notifying law enforcement, 385–386
- investigation, in incident response system, 393
- IOCE (International Organization on Computer Evidence), 384
- ionization fire detector, 282
- IP addresses
 - as part of IP network, 431–432
 - cluster, 681–682
 - mapping domain names to, 473
 - mapping NetBIOS names to, 473
 - private network, 470, 477
 - public network, 470
 - reserved, 476
- IP (Intellectual Property), 367, 368, 374–375
- IPL (Initial Program Load), 463
- IP networks
 - dynamic host configuration protocol on, 474–475
 - node on, 431
- IP Next Generation (IPng), 477
- IPsec (Internet Protocol Security)
 - about, 7, 483
 - as VPN technology
 - AH, 226–227
 - IKE, 225–226
 - in Transport mode, 227
 - in Tunnel mode, 228
 - data in transit protection using, 141
 - in IPv6, 478
- IPS (Intrusion Prevention Systems)
 - about, 124
 - architecture, 127
 - attackers evading, 699
 - attackers using, 129
 - detection mechanisms used by, 125–126
 - monitoring and auditing in, 52
 - types of response to detect attacks, 124
 - vs. intrusion detection systems, 125
- iptables, compensating controls using, 69
- IPv4 (Internet Protocol version 4), 224, 432–433, 471, 475–476
- IPv6 (Internet Protocol version 6), 224, 433, 477–478
- IrDA (Infrared Data Association), 485
- irises of camera lens, in designing physical security, 269
- iris scan, authentication using, 85
- IRTF (Internet Research Task Force), code of ethics, 19
- ISACA (Information Systems Audit and Control Association), 335
- ISAKMP (Internet Security Association and Key Management Protocol), 226

ISC2 Code of Ethics, 18–19, 386
 ISDN (Integrated Services Digital Network), 490
 ISMS (Information Security Management Systems), 333–334
 ISO (International Organization for Standardization)
 15408 series as CC, 348
 27000 series, 17, 333–334
 about, 417
 CC broken into parts by, 349
 on OSI model, 418
 isolation, in transactions, 617
 ISPs (Internet Service Providers), 488
 Israel Defense Force (IDF), launching Stuxnet, 368
 ISRM (Information System Risk Management), 334
 ISSA (Information Systems Security Association), 336
 issue specific policies, 15
 issuing CAs, 212
 ITIL (Information Technology Infrastructure Library), 336
 IT infrastructure
 compatibility of recovered, 550
 documentation of, 549
 inbound and outbound communications, 550
 recovery of data, 551–554
 security standards of, 550
 IT roles, 649–654
 ITSEC (Information Technology Security Evaluation Criteria), 347–348
 IT security standards and guidelines, 345
 IT systems, penetration testing of, 664
 ITU-T (International Telecommunication Union Telecommunication Standardization Sector), 98–100, 215, 490
 IVs (Initialization Vectors), 165, 172

J

Japan, Red and Purple machines, 154
 Java applets, 604
 Java Database Connectivity (JDBC) driver, 611
 Jefferson disk, 154
 job rotation, deterring fraud using, 73, 659
 Just a Bunch Of Disks (JBOD), 674

K

KDC (Key Distribution Center), Kerberos, 75, 94–98
 Kerberized, network operating systems, 94
 Kerberos authentication
 about, 94–98
 authorization vs., 98
 CHAP in, 96, 186
 directory services, 98–100
 federated identity management and, 76
 port used by, 98
 realm in, 95
 weakness of, 98
 Kerckhoffs's principles, 158
 kernel mode, 306, 314–316, 322, 323, 346
 kernel-mode rootkits, 630
 kernel, operating system, 317
 kernel proxy, generation 5 firewall, 469
 key cipher, running, 158
 key clustering, 189
 key (cryptovariable), as component of cryptography, 142
 Key Distribution Center (KDC), Kerberos, 75, 94–98
 key loggers, 699
 key management, 159
 key management process, in target hardening method of designing physical security, 259
 key rotation, 164
 keys, cryptographic
 about, 159
 archive and recovery of, 163
 creation of, 160
 distribution of, 161
 escrow decryption, 163
 length of, 160–161
 lifetime of, 164
 quantities of, 162–163
 secure storage of, 161
 keyspace calculation, 160
 keystroke dynamics, authentication using, 85
 keystroke logger software, 631
 Kiosks, constrained interface in, 119
 Knapsack, 207
 knowledge-based detection, 124
 knowledge base (rule base), 623
 knowledge consistency, 6
 knowledge discovery, 620
 Knowledge level, in knowledge pyramid, 622

knowledge pyramid

knowledge pyramid, 622
known plaintext attack, 184, 236
Kravitz, David, 206

L

L2F (Layer 2 Forwarding), 483
L2TP (Layer 2 Tunneling Protocol), 228–229, 483
Label Edge Router (LER), 482
labeling requirements and technologies, protective controls for, 41
labels, MAC classification, 105–107
Lamarr, Hedy, 495
LAN-based applications, secure channels for, 223–229
LAND (Local Area Network Denial) attack, DoS, 507
LAN (Local Area Network), 141, 486–487
last-mile connectivity, 488
lattice, MAC model, 323–324
laws and regulations
 compliance during disaster recovery with, 542
 defining standard for logging system, 121
 difference between, 373–374
lawsuits
 avoiding discrimination, 15
 impact of negligence on, 12
 reducing risk of, 12–13
 study on employee, 45
Layer 2 Forwarding (L2F), 483
layer-2 MAC address, filtering function of switches, 128
Layer 2 Tunneling Protocol (L2TP), 228–229, 483
Layer 7 function, OSI Model, 421
Layer 7 Simple Mail Transfer Protocol, 421
layered defense model, 246–247, 250
layered security, 10
LDAP (Lightweight Directory Access Protocol), X.500
 Directory Services incorporating, 100
LEAP (Lightweight Extensible Authentication Protocol), Microsoft, 211
leased sites, alternative, 545–546
least privilege, principle of
 Allow access permissions following, 74
 applying, 71, 73, 112
 verifying application of, 104
legal issues, computer crime
 about, 366
 as percentage of total crime, 369
 criminal acts involving, 367–368
global laws
 codified law system, 371
 common law system, 372
 customary law system, 373
 governance of third parties, 382–383
 hybrid law systems, 373
 laws vs. regulations, 373–374
 litigation management, 381
 protecting intellectual property, 374–375
 protecting privacy, 376–382
 religious law system, 373
 software licensing management, 383–384
investigating
 about, 384–385
 evidence gathering, 396–399
 forensic investigations, 399–405
 incident response, 386–395
 notifying law enforcement, 385–386
 motivations for cybercrime, 369
 prosecution of cybercrime, 370
 types of people committing cybercrime, 369
 ways computers are involved in cybercrime, 368
LER (Label Edge Router), 482
level of criticality of MTD, 537
level of risk, quantifying, 9–10
LFSR (Linear Feedback Shift Register)
 about, 174
 RC4 and, 175
liability
 courts identifying, 12
 downstream, 381
life-cycle assurance, 591
lighting, in designing physical security, 267
light timber construction method of facility, as physical control in designing physical security, 255
Lightweight Directory Access Protocol (LDAP), X.500
 Directory Services incorporating, 100
Lightweight Extensible Authentication Protocol (LEAP), Microsoft, 211
likelihood
 countermeasures for reducing or eliminating, 22–23, 31
 of compromise determination, 9
 vs. impact of a qualitative loss event graph, 27
Linear Feedback Shift Register (LFSR)
 about, 174
 RC4 and, 175

Line Printer Daemon (LPD), 481
 link encryption, 209–210
 link state routing protocols, 481
 Linux, DAC implemented in, 112
 litigation
 managing threat of, 381
 reducing risk of, 12–13
 Livingston Enterprises, Inc, development of RADI-US, 116
 LLC (Logical Link Control), 435–436
 lmhosts file (LAN Manager hosts file), 422
 load balanced cluster, 681
 Local Area Network Denial (LAND) attack, DoS, 507
 Local Area Network (LAN), 141, 486–487
 location factors of facility, in CPTED methodology, 253–254
 locking down or disabling, user accounts, 71
 lockout feature
 password, 77
 PIN, 78
 locks, in target hardening method of designing physical security, 258–259
 logging events
 details captured for, 121
 secure disposal of expired data, 124
 logging requirements, software development security, 592–593
 logging system
 auditing and, 120–124, 280
 defining maximum size of log, 134
 laws and regulations defining standard for, 121
 logical controls, 66
 Logical Link Control (LLC), 435–436
 logical memory addressing, 321–322
 logon passwords. *See* passwords
 logrotate daemon, 134
 loose cohesion, 599
 loose coupling, 599
 losses, 8
 lost or stolen authentication devices, users reporting, 84
 low coupling of objects, 599
 low impact risks, 10
 LPD (Line Printer Daemon), 481
 LUC, 207
 Lucas sequences, 207
 Lucifer algorithm, 156, 177

M

M-94 device, 154
 machine language, 595–596
 MAC (Mandatory Access Control)
 about, 105–109
 as nondiscretionary access control, 113
 attributes of, 107
 Biba model applied to, 340–342
 BL model applied to, 339
 government labels used with, 612–614
 in operating system, 323–326
 modes in, 108–109
 permissions provisioning and, 656
 MAC (Media Access Control) address
 about, 435–437
 Message Authentication Codes (MACs) vs., 186
 MAC (Modify/Access/Create), examining times on, 405
 MAC OSX, DAC implemented in, 112
 MACs (Message Authentication Codes)
 about, 185
 digital signatures vs., 186
 Media Access Control (MAC) address vs., 186
 techniques providing, 185–189
 magnetic contact switches, in IDS, 274
 magnetic tape, 555
 MAID (Massive Array of Inactive Disks), 680
 mainframe computers, 463–464
 maintenance hooks, 626
 maintenance phase, in information life cycle, 38
 malicious applications, quarantining, 124–125
 malware
 about, 628–631
 detection mechanisms, 637–638
 in targeted attacks, 699
 manager role, in IT, 650
 managing risk, four methods of, 32–33
 Mandatory Access Control (MAC)
 about, 105–109
 as nondiscretionary access control, 113
 attributes of, 107
 Biba model applied to, 340–342
 BL model applied to, 339
 government labels used with, 612–614
 modes in, 108–109
 operating systems implementing, 107–108, 323–326
 permissions provisioning and, 656
 mandatory vacations, deterring fraud using, 73, 659

Man-in-the-middle (MITM) attack, information theft by

- Man-in-the-middle (MITM) attack, information theft by, 509
- MANs (Metropolitan Area Networks), 488–489
- mantraps, in designing physical security, 266
- A Manuscript on Deciphering Cryptographic Messages (Al-Kindi), 151
- many-to-one (one-to-many) mode, NAT operating in, 472
- Massachusetts Institute of Technology (MIT), development of Kerberos, 94
- Massive Array of Inactive Disks (MAID), 680
- Master File Table (MFT), 688
- master key system, using, 259
- MAU central connection devices, 454
- Mauchly, John, 190
- Maximum Tolerable Downtime (MTD)
 - definitions of, 537
 - determining for business functions, 529, 536–539
 - in strategy for developing disaster recovery plans, 541
- means, in criminal acts, 386
- Mean Time between Failures (MTBF), 673
- Mean Time To Repair (MTTR), 673
- Media Access Control (MAC) address
 - about, 435–436
 - Message Authentication Codes (MACs) vs., 186
- media management, 672, 689–690
- media transmission types
 - about, 443
 - access methods, 459–460
 - cables, 445–449
 - emanations, 443–444
 - encoding data into signals, 450–453
 - networking topologies, 453–459
 - signal degradation, 444
- meeting point leader, fire, 291
- memes, 631
- memory cards, to authenticate users, 81–82
- memory, computer system, 310–311
- memory leak, 320, 626
- memory manager, operating system, 318, 321–323
- memory map, 322
- memory sticks, 307
- Merkle, Ralph, 156, 207
- Message Authentication Codes (MACs)
 - about, 185
 - digital signatures vs., 186
 - Media Access Control (MAC) address vs., 186
 - techniques providing, 185–189
- message digests (hash values)
 - about, 144, 165–168
 - attacks on, 167–168
 - CHAP open-standard and, 91–92
 - hashing algorithms generating, 79
 - output lengths of, 166–167
 - virtual password and, 78
- metadata, 620, 622
- metadirectory, 99
- metamorphic viruses, 629, 637
- Metropolitan Area Networks (MANs), 488–489
- MFT (Master File Table), 688
- MGM Grand Hotel (Las Vegas), fire at, 448
- Microsoft BitLocker Drive Encryption tool
 - as encryption tool, 141
- Microsoft CHAP version 2 (MS-CHAPv2), 484
- Microsoft Lightweight Extensible Authentication Protocol (LEAP), 211
- Microsoft networks, Kerberos realm on, 95
- Microsoft Point-to-Point Tunneling Protocol (PPTP)
 - about, 483
 - as VPN technology, 224
 - data in transit protection using, 141
- Microsoft SDelete utility, 688
- Microsoft Windows
 - ACL implemented in, 110
 - authentication service running on, 75
 - DAC implemented in, 112
- Microsoft Windows Server 2012, MAC/DAC hybrid operating system, 108
- Microsoft Windows XP, automatic teller machines running, 119
- middleware, 327–328, 599, 611
- MIME (Multipurpose Internet Mail Extensions), 480
- MIME (Multipurpose Internet Message Extensions), 234
- MIMO (Multiple Input/Multiple Output), 503
- mirror port, sensors attaching to layer-2 MAC switches using, 128
- misdemeanors, 372
- mitigating risk, countermeasure of, 22–23, 32
- MIT (Massachusetts Institute of Technology), development of Kerberos, 94
- MITM (Man-in-the-middle) attack, information theft by, 509
- Mitre Corporation, tracking CVEs, 582
- MixColumns function, 178
- mobile code, 604
- models, database, 607–610

Modify/Access/Create (MAC), examining times on, 405
 modular objects, 597
 monitoring
 about, 51
 auditing and, 51–54
 incident responses, 389–391
 user activity, 71
 monitoring policy
 employee privacy issues and, 379–380
 informing prospective employees of, 46–47
 monitoring station for CCTV system, in designing physical security, 270
 monitoring trends, 76
 monoalphabetic ciphers, 149
 Montreal Accord, 286–287
 Moore’s Law, 147–148
 Morris, Robert, 156
 Morris Worm, 156, 386, 630
 Most Recently Used (MRU), 701
 motherboard, 307
 motion-triggered lighting, in designing physical security, 267
 motive, opportunity and means, in criminal acts, 386
 mounting of camera, in designing physical security, 270
 MPLS (Multi-Protocol Label Switching), 380, 482
 MRU (Most Recently Used), 701
 MS-CHAP, 186
 MS-CHAPv2 (Microsoft CHAP version 2), 484
 MTBF (Mean Time between Failures), 673
 MTD (Maximum Tolerable Downtime)
 definitions of, 537
 determining for business functions, 529, 536–539
 in strategy for developing disaster recovery plans, 541
 Multicast mode, 458
 multi-factor system authentication, 69
 multi-factor (two-factor) authentication, 89–90
 multihomed firewalls, 469–470
 multi-level security mode, MAC, 109
 multi-level security mode, MAC model, 326
 Multiple Input/Multiple Output (MIMO), 503
 multiplexing, 309, 456–457
 multiprocessing computer system, 309
 multiprogramming feature, operating systems, 316
 Multi-Protocol Label Switching (MPLS), 380, 482
 Multipurpose Internet Mail Extensions (MIME), 480
 multitasking feature, operating systems, 317–318
 multithreading concept, operating systems, 317

mutual authentication
 about, 90–91
 Kerberos providing, 98
 zero knowledge proof, 91–93

N

name resolution, 473–474
 namespace, as realm boundary, 95
 nanosecond, 307
 NAS (Network Access Server), RADIUS servers and, 116
 NAS (Network Attached Storage) systems, 680
 National Commission on Fraudulent Financial Reporting, 335
 National Computer Security Center, US, IT security standards and guidelines by, 345
 National Electric Code (NEC), on standards for plenum-rated cabling, 448
 National Fire Protection Agency (NFPA)
 on standards for plenum-rated cabling, 448
 standard for portable fire extinguishers, 290–291
 National Institute of Standards and Technology (NIST)
 C&A defined by, 344
 CMAC recommended by, 189
 DSS introduced by, 206
 list of SP 800 series documents, 355–360
 Lucifer algorithm and, 156
 on preparing clone disk for forensic investigations, 402
 RC6 recommended by, 179
 recommendation on ECC key, 206
 recommended cryptoperiod of cryptographic keys, 164
 recommended minimum symmetric key length, 161
 SP 800 series, 18, 336
 tracking CVEs, 582
 National Oceanic Atmospheric Administration (NOAA), 29
 National Security Agency (NSA)
 approval of AES, 178
 DSS patented by, 206
 ECC approval for protecting secret or classified information, 206
 hashing algorithms at, 156
 IT security standards and guidelines, 345
 ran project codenamed Tempest, 444
 NAT (Network Address Translation), 227, 467, 471–472

natural disasters, understanding potential of

natural disasters, understanding potential of, 253

natural threats

about, 9, 247, 539

in performing risk assessment, 29

nature of loss, 8

nature of the key

abstracting, 145

adaptive attack focusing on, 237

ciphertext showing patterns of, 180

hiding within ciphertext, 237

in Atbash cipher, 149

in Caesar or Shift cipher, 151

in ciphertext, 165

of running key cipher, 158

Nazi Germany, using Enigma machine, 154

NDAs (Nondisclosure Agreements)

administrative controls as, 104

implementing trade secrets with, 375

informing prospective employees of, 46–47

in pen testing agreement, 668

NDS (Novell Directory Services), X.500 Directory Services and, 99

NEC (National Electric Code), on standards for plenum-rated cabling, 448

negligence

components of avoiding, 12–13

definition of, 12

impact on lawsuits of, 12

monitoring and auditing to avoid, 52

NetBIOS name, 422, 473

Network Access Server (NAS), RADIUS servers and, 116

network address translation, 471–472

Network Address Translation (NAT), 227, 467, 471–472

Network Attached Storage (NAS) systems, 680

network attacks, understanding, 416

network-based IDS (NIDS or HIPS), 125, 128

network classification, 457–458

network connections

hazardous, 7

hazardous network connections, 141

network database model, 608

network devices

bastion host/hardened systems, 465–467

client/endpoint systems

about, 464–465

remote access by, 465

dynamic host configuration protocol, 474–475

firewalls, 467–471

mainframe computers, 463–464

name resolution, 473–474

network address translation, 471–472

virtual private networks, 475

within OSI model, 460–463

Network File System (NFS), 481, 680

networking topologies, 453–459

Network Interface Cards (NICs)

about, 463

driver, 438

hardware guard and, 119

in hardware wireless networks, 493–494

manufacturer in MAC address, 436

on network sensors, 128

wireless, 90

network isolation (air gapping), as compensating control, 69

Network layer of OSI Model, 431–435, 462

networks. *See also* wireless networks

attacks on

about, 505

DDoS, 507–508

DoS, 506–507

for information theft, 508–511

types of, 505

classful, 476–477

local area, 486–487

MANs (Metropolitan Area Networks), 488–489

PBX system, 491

personal area, 485–486

testing effectiveness of perimeter, 128

voice over Internet protocol, 491–492

wide area, 489–491

network sensors

in and outside external firewall, 128

in promiscuous mode, 128

mechanisms to defeat IDS, 127

Network Time Protocol (NTP), 390

using in resolving time synchronization issues, 98

NFPA (National Fire Protection Agency)

on standards for plenum-rated cabling, 448

standard for portable fire extinguishers, 290–291

NFS (Network File System), 481, 680

NICs (Network Interface Cards)

about, 463

driver, 438

hardware guard and, 119

in hardware wireless networks, 493–494

- manufacturer in MAC address, 436
 - on network sensors, 128
 - wireless, 90
 - NIDS or NIPS (network-based IDS), 125, 128
 - NIST (National Institute of Standards and Technology)
 - C&A defined by, 344
 - CMAC recommended by, 189
 - DSS introduced by, 206
 - list of SP 800 series documents, 355–360
 - Lucifer algorithm and, 156
 - on preparing clone disk for forensic investigations, 402
 - RC6 recommended by, 179
 - recommendation on ECC key, 206
 - recommended cryptoperiod of cryptographic keys, 164
 - recommended minimum symmetric key length, 161
 - SP 800 series, 18, 336–337
 - tracking CVEs, 582
 - NOAA (National Oceanic Atmospheric Administration), 29
 - No Access default
 - allow permissions applied to, 103
 - using DAC, 110
 - nodes
 - about, 453
 - on Internet, 458
 - on IP network, 431
 - no disclosure double blind testing, 667
 - no expectation of privacy, employee awareness of, 380
 - noise in signal transmission, 262, 278
 - Noise level, in knowledge pyramid, 622
 - nonce, as nonsecret variable, 165
 - Nonce Sum (NS), TCP segment, 426
 - noncompete agreements, informing prospective employees of, 46–47
 - nondeterministic media access system, 459–460
 - Nondisclosure Agreements (NDAs)
 - administrative controls as, 104
 - implementing trade secrets with, 375
 - informing prospective employees of, 46–47
 - in pen testing agreement, 668
 - nondiscretionary access control vs. discretionary access control, 113
 - nonflammable gases, for fire suppression, 290
 - noninterference model, computer systems, 339
 - nonpersistent cross-site scripting attack, 632–633
 - nonpromiscuous mode, NIC operating in, 128
 - nonrepudiation
 - asymmetric key cryptography providing, 140
 - in cryptosystem, 143
 - provided by digital signatures, 192
 - symmetric keys and, 170–171
 - nonsecret variable, IVs as, 165
 - nonvolatile memory, 310
 - normalization, data, 620
 - notification, in incident response system, 392
 - Novell Directory Services (NDS), X.500 Directory Services and, 99
 - NSA (National Security Agency)
 - approval of AES, 178
 - DSS patented by, 206
 - ECC approval for protecting secret or classified information, 206
 - hashing algorithms at, 156
 - IT security standards and guidelines, 345
 - ran project codenamed Tempest, 444
 - NS (Nonce Sum), TCP segment, 426
 - NTP (Network Time Protocol)
 - about, 390
 - using in resolving time synchronization issues, 98
- ## O
- Oakley Key Determination Protocol, 226
 - OASIS (Organization for the Advancement of Structured Information Standards), development of SAML, 603
 - object, in object-oriented programming
 - about, 597
 - encapsulation of, 598–599
 - high cohesiveness of, 599
 - inheritance feature of, 598
 - instantiating class into, 598
 - low coupling of, 599
 - objectives and controls, security, 5–7
 - objectives and goals, presented in plans, 5
 - Object Linking and Embedding Database (OLE DB), 611
 - Object Linking and Embedding (OLE), 600
 - object-oriented database model, 610
 - Object-Oriented Programming (OOP), 597–599
 - Object Request Broker (ORB) standard, 601
 - object reuse, 689–690
 - objects statement, in trusted path, 64

Occupational Safety and Health Administration (OSHA)

- Occupational Safety and Health Administration (OSHA), 372
- OCSP (Online Certificate Status Protocol), accessing CRL using, 221
- ODBC (Open Database Connectivity) drivers, 99, 611–612, 620
- OFB (Output Feedback mode), as mode of symmetric key block ciphers, 182
- OFDM (Orthogonal Frequency Division Multiplexing), 496–497
- OLE DB (Object Linking and Embedding Database), 611
- OLE (Object Linking and Embedding), 600
- OLTP (Online Transaction Processing), 617
- one round of cryptographic processing, 175–176
- one-time pad (Vernam cipher), 154
- one-time password, 78, 83
- one-to-many identification method, in biometric system, 87–88
- one-to-many (many-to-one) mode, NAT operating in, 472
- one-to-one mode, NAT operating in, 472
- one-to-one verification method, in biometric system, 87
- online analytical processing, 617
- Online Certificate Status Protocol (OCSP), accessing CRL using, 221
- Online Transaction Processing (OLTP), 617
- online UPS, 276
- OOP (Object-Oriented Programming), 597–599
- Open Database Connectivity (ODBC) drivers, 99, 611–612, 620
- OpenPGP, 222
- Open Shortest Path First (OSPF), 482
- open source software, 586
- Open Systems Interconnection (OSI) Model
 - about, 417
 - network devices within, 460–463
 - seven layers of
 - about, 418–421
 - Application layer, 421–422, 463
 - Data Link layer, 435–440, 461–462
 - Network layer, 431–435, 462
 - Physical layer, 441, 461
 - Presentation layer, 422–423
 - Session layer, 423–424
 - Transport layer, 424–430
 - understanding network attacks and, 416
- Open Web Application Security Project (OWASP), 603–604
- operating systems
 - about, 314–316
 - buffer overflow attack, 320–321
 - implementing MAC, 107–108
 - in safe mode, 346
 - MAC model within, 323–326
 - memory manager, 321–323
 - multiprogramming feature, 316
 - multitasking feature, 317–318
 - multithreading concept, 317
 - processes, 318–320
- operational assurance, in software development, 591
- operational goals, definition of, 5
- operation and maintenance, in software development
 - life cycle, 582–583
- operations security
 - about, 647
 - activities of operations
 - about, 648–649
 - availability of operation systems, 655–656
 - fraud protection, 657–661
 - IT roles, 649–654
 - remote access connections, 653–654
 - remote administration, 654
 - user provisioning life cycle, 656–657
 - vulnerability assessments, 661–670
 - attacks on operations
 - about, 691
 - anatomy of targeted attacks, 693–701
 - common attacks and losses, 692
 - preventive measures, 691–692
 - data management
 - about, 671
 - classifying data, 39–45, 671
 - fax security, 690–691
 - maintaining systems supporting data, 673–687
 - media management, 672
 - object reuse, 689–690
 - retention of data, 687–688
 - secure deletion of hard disk drive data, 688–689
 - secure destruction policy, 689–690
- opportunistic attacks, 693
- opportunity, in criminal acts, 386
- Orange Book, 345, 347, 591
- ORB (Object Request Broker) standard, 601
- organizational policy, 15

organization chart, in implementing security program, 36–37

Organization for the Advancement of Structured Information Standards (OASIS), 603

Orthogonal Frequency Division Multiplexing (OFDM), 496–497

OSHA (Occupational Safety and Health Administration), 372

OSI (Open Systems Interconnection) Model

- about, 417
- network devices within, 460–463
- seven layers of
 - about, 418–421
 - Application layer, 421–422, 463
 - Data Link layer, 435–440, 461–462
 - Network layer, 431–435, 462
 - Physical layer, 441, 461
 - Presentation layer, 422–423
 - Session layer, 423–424
 - Transport layer, 424–430
- understanding network attacks and, 416

OSPF (Open Shortest Path First), 482

out-of-band communications channel, 392

Output Feedback mode (OFB), as mode of symmetric key block ciphers, 182

overt (intended communications channel), 627

OWASP (Open Web Application Security Project), 603–604

P

P2P (Peer-To-Peer) networks, 329

PaaS (Platform as a Service), 331, 465

packet, 431–435

packet filter firewall, 462, 628

Packet filter, generation 1 firewall, 468

packet-switched vs. circuit-switched networks, 455–456

packet transmission modes, 458–459

PAC (Privileged Attribute Certificate), SESAME calling, 100

padded cells, evidence gathered from, 129–130

page fault events, virtual memory manager recognizing, 322

palmprint, authentication using, 84

PAN (Personal Area Network), 485–486

PAP (Password Authentication Protocol), 484

parallel processing facilities, 544

parallel processing vs. distributed systems, 329

parallel testing, 567

parity, RAID, 678–679

Parker, Donn B., 248

partial disclosure testing, 667

PAS (Privileged Attribute Server), in SESAME, 100

pass-by traffic at location of facility, 254

passive reconnaissance, in targeted attacks, 694–695

passphrase

- about, 78
- vs. virtual password, 78

passwd file, accessing, 698

Password Authentication Protocol (PAP), 484

password generators, 77

passwords

- as authentication mechanism, 77
- attacks on, 79–81
- cognitive, 78
- cracking tool for, 512
- dictionary attack on, 144
- hybrid attack on, 144
- one-time, 78, 83
- portable device, 271
- rainbow attack on, 80, 144
- resetting, 79
- social engineering attack on, 145
- standard for complexity of, 77
- using with device-related mechanisms, 84
- virtual, 78
- zero knowledge proof in mutual authentication, 91

password synchronization, systems supporting own form of user database and, 100

patents, 374

PAT (Port Address Translation), 471

pattern detection attack, 145

patterns and trends, identifying, 621

Payment Card Industry Data Security Standard (PCI-DSS), 17, 52–53, 378–379, 582, 664

Payment Card Industry (PCI), 233, 349

PBX phone system, attacks through, 513

PBX (Private Branch Exchange), 491

PCI-DSS (Payment Card Industry Data Security Standard), 17, 52–53, 378–379, 582, 664

PCI (Payment Card Industry), 233, 349

PC (Printed Circuit) board, 307

PDCA (Plan-Do-Check-Act) model, 333

PEAP (Protected Extensible Authentication Protocol), 211, 224

Peer-To-Peer (P2P) networks

- Peer-To-Peer (P2P) networks, 329
- penetration testing, 124, 662–669
- pen testing, 662–669
- percent encoding, 636
- Perimeter Intrusion Detection and Assessment System (PIDAS), 261
- perimeter network, testing effectiveness of, 128
- perimeter security, goals when implementing, 250–252
- peripherals, computer system, 312
- permissions. *See also* access controls
 - allow, applied to No Access default, 103
 - defining privileges of users, 103
 - explicit, 114
- permissions provisioning, 656
- permutation ciphers (transposition ciphers)
 - about, 145
 - AES performing, 178
 - Blowfish performing, 179
 - IDEA performing, 175
 - Scytale performing, 150
 - symmetric key algorithms performing, 171
 - Twofish performing, 179
- persistent cookies, 637
- persistent cross-site scripting attack, 633–634
- Personal Area Network (PAN), 485–486
- personal firewall (host-based firewall), 467
- Personal Identification Numbers (PINs)
 - using in authenticating users, 78
 - using with device-related mechanisms, 84
- personally identifiable information (PII), 367–368, 508
- PGP (Pretty Good Privacy), 192, 221–223, 688
- Phase-Shift Keying (PSK), 451–452
- phone systems, attacks on, 513
- photoelectric fire detector, 282–283
- photoelectric sensors, in IDS, 273
- photo ID badges, to authenticate users, 81
- phreakers, 513
- physical access controls
 - about, 11, 66
 - functional security objectives and, 70
 - in designing physical security, 265–266
 - MAC and, 108
 - to defend against fraud, 660
- physical assets, reviewing list of company-owned, 48
- physical controls of facility, in designing physical security, 254–256
- Physical layer of OSI Model, 441, 461
- physical security
 - as first line of defense, 247
 - auditing and logging system, 280
 - building layered defense, 246–247
 - designing
 - about, 249–252
 - avoiding piggybacking, 265
 - building layered defense, 250
 - CCTV cameras in, 267–270
 - CPTED methodology in, 252–256
 - doors, 258
 - emanations protection, 261–263
 - fail safe and fail secure in, 266
 - failure recovery considerations, 275–279
 - fences, 259–261
 - full wall vs. partition, 257
 - guard dogs as part of, 265
 - HVAC systems in, 274–275
 - intrusion detection and prevention systems in, 272–274. *See also* Intrusion Detection Systems (IDS)
 - key management process, 259
 - lighting in, 267
 - locks, 258–259
 - physical access controls, 265–266
 - protecting against piggybacking method, 265
 - securing portable devices, 270–272
 - security guard considerations, 263–264
 - signage in, 266–267
 - target hardening method, 257–272
 - window design, 257
 - fire and
 - about, 281
 - classes of fires, 283–284
 - fire extinguishers, 288–291
 - four legs of fire, 281–282
 - plan and drill, 291–292
 - sprinkler systems, 284–286
 - suppression agents, 286–290
 - types of detectors, 282–283
 - goals when implementing perimeter security, 250–252
 - liability of physical design, 248
 - periodic walkthroughs and inspections, 279–280
 - planning design, 246–248
 - physical segmentation, on computer systems, 313
 - physical threats, 247–248

- pick-resistant locks, in target hardening method of designing physical security, 258
- PIDAS (Perimeter Intrusion Detection and Assessment System), 261
- PID (Process Identifier), 318
- piggybacking (tailgating) method, 265
- PII (personally identifiable information), 367–368, 508
- pillage phase, in targeted attacks, 701
- Ping, DoS attacks with, 506
- pinned path requirements, protocols supporting, 380
- PINs (Personal Identification Numbers)
 - using in authenticating users, 78
 - using with device-related mechanisms, 84
- pipelining, CPU chips supporting, 309
- pirated software, 384
- pivot and attack phase, in targeted attacks, 701
- PKI (Public Key Infrastructure)
 - about, 210–211
 - as standard for security, 192
 - certificate repository, 220
 - Certification Authorities in, 211
 - digital certificates from, 83, 211
 - fraud protection in archiving, 659
 - ITU-T and, 100
 - PGP vs., 222
 - RAs authenticating user in, 213–214
 - trusting certification authority or, 214–215
 - X.509 digital certificate, 215–216
- PK (Primary Key), 613
- Plain Old Telephone System (POTS), 455
- plaintext attacks
 - chosen, 237
 - known, 184, 236
- plaintext message
 - as component of cryptography, 142
 - encryption converts into ciphertext, 142
 - known plaintext attack, 184
 - using in XOR function, 173
- Plan-Do-Check-Act (PDCA) model, 333
- planning budget, defining, 532–533
- planning project leader, defining, 530
- planning team, defining, 531
- Platform as a Service (PaaS), 50, 331, 465
- platform-independent technology, 601
- PLD (Programmable Logic Devices), 310
- plenum-rated cabling, 448
- Point-to-Point Protocol (PPP), 228, 489
- Point-to-Point Tunneling Protocol (PPTP), Microsoft
 - about, 483
 - as VPN technology, 224
 - data in transit protection using, 141
- poison reverse advertisement, 482
- policies and frameworks, 14–15
- policies of organization, 332
- policy CAs, 212
- policy documents
 - about, 15
 - awareness of content of, 20
 - certification and accreditation of, 19–20
 - conveying ethical standards, 18–19
 - for fraud protection, 658
 - project of risk management and, 21
 - provisioning process in, 104
 - revisions, updates, and change control of, 21
 - sources for developing, 16–18
- polling method, 459
- polyalphabetic ciphers, 149, 153
- polyinstantiation, 341–342, 612–614
- polymorphic viruses, 629, 637
- polymorphism, in object-oriented programming, 598
- Polyvinyl Chloride (PVC) plastic, 447–448, 490
- POP3, 480
- portable devices, securing, 270–272
- Port Address Translation (PAT), 471
- port forwarding mode, NAT operating in, 472
- port numbers, protocols and, 428–430
- ports
 - Kerberos, 98
 - Network Time Protocol, 98
- port scanning, 696
- positive air pressure, ventilation systems producing, 275
- POTS (Plain Old Telephone System), 455
- power generators, 548
- PPP (Point-to-Point Protocol), 228, 489
- PPTP (Point-to-Point Tunneling Protocol), Microsoft
 - about, 483
 - as VPN technology, 224
 - data in transit protection using, 141
- practice restore of the data, 558–559
- pre-action sprinkler systems, 285
- precedents, verdicts based on, 370
- preemptive multitasking feature, operating systems, 317
- preponderance of the evidence, burden of proof, 372

presentation in court, in evidence life cycle

- presentation in court, in evidence life cycle, 397
- Presentation layer of OSI Model, 422–423
- preservation of (crime) scene, in evidence life cycle, 396
- pressure mats, in IDS, 274
- Pretty Good Privacy (PGP), 192, 221–223, 688
- preventive controls, 67, 541
- Primary Key (PK), 613
- principle of least privilege
 - Allow access permissions following, 74
 - applying, 71, 73, 112
 - malware and, 628
 - privilege escalation in targeted attacks and, 697
 - provisioning requests and, 656
 - verifying application of, 104
- Printed Circuit (PC) board, 307
- privacy, protecting, 7, 376–382
- Private Branch Exchange (PBX), 491
- private IP addresses, 470, 477
- private key cryptography, asymmetric
 - key distribution in, 161
 - mathematics and, 202
 - public key vs., 191
 - quantities of keys in, 162–163
- private key, encrypted on CAC, 217
- Privileged Attribute Certificate (PAC), SESAME calling, 100
- Privileged Attribute Server (PAS), in SESAME, 100
- privileged users assessment, 662
- privilege escalation, in targeted attacks, 697–698
- privilege review, assigned, 104
- privilege revocation processes, principle of least privilege and, 105
- privileges
 - applying principle of least, 71, 73, 112. *See also* principle of least privilege
 - defining users, 103
 - vs. trust, 96
- PRNG (Pseudo-Random Number Generator), 172
- procedural programming, 597
- procedure documents, 16
- processes, operating system, 318–320
- processes sharing data
 - on multiple computers, 600–601
 - on single computer, 600
- Process Identifier (PID), 318
- processors, 595
- process states, operating system, 318–319
- Programmable Logic Devices (PLD), 310
- programming concepts
 - about, 595–596
 - distributed computing, 599–604
 - generations of programming languages, 596–597
 - object-oriented programming, 597–599
- Project Athena, 94
- project deliverables
 - defining scope of project, 531
 - scheduling, 532–533
- project initiation, in software development life cycle, 580
- project-planning models, software, 588–590
- promiscuous mode, NIC operating in, 128
- prosecution of cybercrime, 370
- Protected Extensible Authentication Protocol (PEAP), 211, 224
- protected memory, operating system, 318
- protection of data
 - data in use and, 7
 - privacy issues, 376–382
- protection of integrity, 6
- protection of verification, 6
- Protection Profile, as component in evaluation and certification process, 348
- protective controls, in implementing security program, 41–43
- protocols
 - about, 475
 - authentication, 484
 - commonly used, 479–481
 - Internet Protocol version 4, 475–477
 - Internet Protocol version 6, 477–478
 - port numbers and, 428–430
 - routing, 481–482
 - TCP/IP Protocol suite, 478–479
 - virtual private network, 482–484
- provisioning
 - about, 104
 - life cycle
 - about, 70–71
 - managing fraud and, 72–73
 - provisioning request document, 656
- proximate causation, 13
- proximity detectors, in IDS, 273
- Proxy server, generation 2 firewall, 468–469
- prudent person rule, 13, 333
- pseudo-randomness, 174

Pseudo-Random Number Generator (PRNG), 172
 PSH (Push), TCP segment, 426
 PSK (Phase-Shift Keying), 451–452
 P@ssword, using as input, 167
 PSTN (Public Switched Telephone Network), 455, 489
 public IP addresses, 470
 public key, bound to digital certificate, 217
 public key cryptography, asymmetric
 key distribution in, 161
 key management in, 159
 keys in
 quantities of, 162–163
 length of, 160–161
 mathematics and, 202
 private key vs., 191
 Public Key Infrastructure (PKI)
 about, 210–211
 as standard for security, 192
 certificate repository, 220
 certificate revocation, 220–221
 Certification Authorities in, 211
 digital certificates from, 83, 211
 fraud protection in archiving, 659
 ITU-T and, 100
 PGP vs., 222
 RAs authenticating user in, 213–214
 trusting certification authority or, 214–215
 X.509 digital certificate, 215–220
 Public Switched Telephone Network (PSTN), 455, 489
 Purple and Red machines (Japanese), 154
 Purple Penelope operating system, MAC implementation in, 107
 Push (PSH), TCP segment, 426
 PVC (Polyvinyl Chloride) plastic, 447–448, 490

Q

QA/developer loop, 584
 Quadrature Amplitude Modulation (QAM), 452
 qualitative loss event, quantifying, 27
 qualitative values, 26
 Quality Assurance/developer loop, 584
 Quality Assurance (QA) review, in software development, 581
 Quality of Service (QoS), 478
 quantitative values, 25–28

quarantine
 indications of Trojan and malicious application to, 125
 malicious or suspicious executable processes to, 124

R

Radio Frequency Identification Devices (RFID), monitoring physical environment using, 53–54
 Radio Frequency Interference (RFI), 262, 278, 444
 Radio Frequency (RF)
 transponders emitting signals, 82
 waves, 449
 white noise, 263, 443
 RADIUS (Remote Authentication Dial-In User Services), 116–117
 RAD (Rapid Application Development), software project-planning models, 590
 RAID (Redundant Array of Independent Disks)
 about, 553, 674–677
 parity, 678–679
 rainbow attack, on passwords, 80, 144
 RAIT (Redundant Array of Independent Tapes), 679
 RAM (Random Access Memory)
 about, 310
 location of, 307
 memory leaks depleting, 320
 memory manager allocating, 318
 trusted recovery attempt and, 346
 virtual memory manager and, 322–323
 randomness, achieving, 174
 ransomware, 631
 Rapid Application Development (RAD), software project-planning models, 590
 RA (Registration Authority)
 authenticating user in PKI system, 213–214
 digital certificates using, 83
 RARP (Reverse Address Resolution Protocol), 475
 raw data, 622
 RBAC (Role-Based Access Control), 113–114, 323
 RC4 (Rivest Cipher 4), 157, 175
 RC5 (Rivest Cipher 5), 179
 RC6 (Rivest Cipher 6), 179
 reader devices, for contactless cards, 82
 Read-Only Memory (ROM), 310
 real evidence, 398
 realm (domain), Kerberos, 95

real-time

- real-time
 - backup, 552
 - copies, 551
 - monitoring, 53, 123
- Real-Time Protocol (RTP), 492
- reciprocal agreements, 546–547
- reconstitution guidelines, of BCP, 560–561
- record locking, 614
- records (tuple), 605
- recovery, in incident response system, 395
- recovery of data, factors in, 551–554
- recovery of personnel, 559
- Recovery Point Objective (RPO), 551, 683
- Recovery Time Objective (RTO), 552
- Red and Purple machines (Japanese), 154
- Red Book, 347
- redirector, 422
- Reduced Instruction Set Code (RISC), 308–309
- Reduced Instruction Set Computing (RISC) instruction set, 595
- redundancy
 - data, 673
 - in recovery plan, 655–656
 - server, 681–682
- Redundant Array of Independent Disks (RAID)
 - about, 553, 674–677
 - parity, 678–679
- Redundant Array of Independent Tapes (RAIT), 679
- referential integrity validation, 609
- registered and approved trademarks, 375
- Registration Authority (RA)
 - authenticating user in PKI system, 213–214
 - digital certificates using, 83
- regression testing, in software development, 583, 591
- regulations and laws
 - compliance during disaster recovery with, 542
 - defining standard for logging system, 121
 - difference between, 373–374
- regulatory law, 372
- relational database model, 608
- relevance of evidence, 397
- religious law system, 373
- remnants
 - elimination of, 404
 - elimination of hard disk, 401, 688
- remnants, operating system data, 319
- remote access, by client/endpoint systems, 465
- remote access connections, 653–654
- remote administration, 654
- remote attestation, TRM chips featuring, 313
- Remote Authentication Dial-In User Services (RADIUS), 116–117
- Remote Procedure Calls (RPC), 601
- removable media, 555
- repeater, 444
- replay attack, on passwords, 80
- reporting, in incident response system, 395
- reporting penetration testing activities, 668–669
- Reset (RST), TCP segment, 426
- resident viruses, 629
- residual risk, in ALE calculation, 30
- resolver, 434
- responding to intruders, 250, 252
- response system, IDS, 125
- résumés, verifying, 46
- retention of data, 687–688
- retention period requirements, protective controls for, 42
- retina scan, authentication using, 85
- return the evidence to the victim, in evidence life cycle, 397
- Reverse Address Resolution Protocol (RARP), 475
- revisions, of policy documents, 21
- rewritable optical disks, reusing, 689
- RFID (Radio Frequency Identification Devices), monitoring physical environment using, 53–54
- RFI (Radio Frequency Interference), 262, 278, 444
- RF (Radio Frequency)
 - transponders emitting signals, 82
 - waves, 449
 - white noise, 263, 443
- RG-8/U, Thicknet (10BASE5 coax), 446
- rights, defining privileges of users, 103
- Rijmen, Vincent, 157
- Rijndael algorithm, 157
- ring architecture, 315–316
- RIP (Routing Information Protocol), 481
- RIPv2, 481
- RISC (Reduced Instruction Set Code), 308–309
- RISC (Reduced Instruction Set Computing) instruction set, 595
- risk assessment
 - about, 21–23
 - completing, 33–34
 - implementing security program, 34. *See also* implementing security program

- performing
 - assigning values to assets, 25–28
 - calculating annualized loss expectancy, 29–31
 - classifying assets, 28
 - four methods of managing risk, 31–32
 - identifying cost-effective countermeasures, 31–32
 - identifying threats, 28–29
 - inventory assets, 24–25
 - managing speculation and uncertainty, 33
 - using automated tools for, 24
- risk avoidance, 33
- risk management
 - about, 21–23
 - information security and, 2–5
- risk management flowchart, 11
- risk management project, starting, 23–24
- risk mitigation, 32
- risk modeling, 8–10
- risks
 - about, 9
 - low impact, 10
 - protecting organization against, 10
 - quantifying, 10
- risk transference, 33
- Rivest Cipher 4 (RC4), 157, 175
- Rivest Cipher 5 (RC5), 179
- Rivest Cipher 6 (RC6), 179
- Rivest, Ron, 205
- roaming, 497–498
- Role-Based Access Control (RBAC)
 - about, 113–114
 - MAC vs., 323
- rollback transaction processing concurrency control, 615
- ROM (Read-Only Memory), 310
- root CAs, 211
- rootkits, 630, 699
- round function, 159
- rounds of cryptographic processing
 - AES, 178
 - Blowfish, 179
 - IDEA, 175
 - one, 175–176
 - Twofish, 179
- route poisoning, information theft by, 510
- routers, 434, 462
- routing, 434–435

- Routing Information Protocol (RIP), 481

- routing protocols
 - dynamic, 462, 481–482
 - static, 481

- RPC (Remote Procedure Calls), 601

- RPO (Recovery Point Objective), 551, 683

- RSA asymmetric key algorithm, 205

- RST (Reset), TCP segment, 426

- RTO (Recovery Time Objective), 552

- RTP (Real-Time Protocol), 492

- rule-based access control, 114–115

- rule base (knowledge base), 623

- rules, enforcement of, 14–15

S

- SaaS (Software as a Service), 465

- SaaS (Storage as a Service), 679

- SABSA (Sherwood Applied Business Security Architecture), 17, 334

- safe mode, computer system, 346

- safety measures, implementing and maintaining, 12

- safety warden, fire, 291

- sag, electrical, 278

- salami attack, 367

- salt, as nonsecret variable, 165

- SAML (Security Assertion Markup Language), 102, 603

- SAM (Security Accounts Manager) database, 698

- SAN (Storage Area Networks), 679–680

- Sarbanes-Oxley Act (SOX) of 2002, 17, 351, 372, 377

- satellite communications, 491

- S-box (Substitution box)

- about, 172

- function, 173, 178

- symmetric key block ciphers using, 175–176

- scalar processing, CPU chip supporting, 309

- scheduling project deliverables, 532–533

- schema of database, 99, 606

- Schneier, Bruce, 179

- Schnorr, Claus, 206

- SCOMP operating system, MAC implementation in, 107

- scope creep, 23, 531

- scope of assessment, for starting risk management project, 23–24

- screened hosts, 469

- screened subnet firewall, 69

script kiddies

- script kiddies, 369
- scripts, 604
- scrubbing logs, 123, 700
- Scytale cipher, 150
- SDLC (Software Development Life Cycle), 579–586
- SDLC (Synchronous Data Link Control), 463
- sealing messages
 - sending to multiple recipients when, 197
 - using asymmetric key algorithms, 195–201
- sealing messages, using symmetric key algorithms, 185, 189
- secondary authentication mechanisms, computer hardware and, 313
- secondary power supplies, in designing physical security, 275–277
- SECRET clearance label, 612–614
- secret keys, 164
- SEC (Securities and Exchange Commission), SOX enacted due to reporting to, 351
- sectors, hard disk, 403
- secure code review, 580
- secure deletion of hard disk drive data, 688–689
- secure destruction policy, 690
- secure disposal
 - of the expired data, 124
 - protective controls for, 42
- Secure Electronic Transaction (SET), 233–234, 379
- Secure European System for Applications in a Multivendor Environment (SESAME), 100–101
- Secure FTP (SFTP), 224, 233, 479
- Secure Hypertext Transfer Protocol (S-HTTP), 232
- secure key distribution
 - in cryptosystem, 143
 - symmetric keys and, 171
- Secure Key Exchange Mechanism (SKEME), 226
- Secure Multipurpose Internet Message Extensions (S/MIME), 234, 480
- Secure Shell (SSH), 223, 479, 483
- Secure Sockets Layer (SSL)
 - about, 7, 483
 - certificate, 230
 - data in transit protection over Internet, 141
 - vulnerability in, 696
 - web-based authentication and, 101
- Secure Socket Tunneling Protocol (SSTP), 229, 484
- secure storage, in evidence life cycle, 397
- Securities and Exchange Commission (SEC), SOX enacted due to reporting to, 351
- security
 - role of professional, 4
 - targets for providing, 5
- Security Accounts Manager (SAM) database, 698
- security architecture
 - about, 303
 - application architecture, 326–332
 - architecture of computer system, operating system, and applications, 306
 - C&A frameworks, 344–349
 - chart of major components of computer system, 312
 - computer hardware and, 307–314
 - frameworks for security
 - about, 332–333
 - COBIT, 335
 - COSO, 335
 - GAISP, 336
 - ISO 27000 series, 333–334
 - ITL, 336
 - NIST SP 800 series, 336–337
 - Zachman Framework, 334
- identifying architectural boundaries, 304–305
- legal and regulatory compliance, 349–352
- operating systems
 - about, 314–316
 - buffer overflow attack, 320–321
 - MAC model within, 323–326
 - memory manager, 321–323
 - multiprogramming feature, 316
 - multitasking feature, 317
 - multithreading concept, 317
 - processes, 318–320
 - security models, 337–343
- Security as a Service (SECaaS), 50
- Security Assertion Markup Language (SAML), 102, 602
- security assessments
 - as auditing functions, 124
 - penetration testing relationship to, 663
 - vulnerability assessments relationship to, 661
- security awareness training program, 20, 44, 49, 658
- security controls
 - implementing information system, 304–305
 - monitoring, testing and auditing, 279–280
- security domain, formation of, 95

- security guards
 - considerations in designing physical security, 263–264
 - performing periodic walkthroughs and inspections, 279–280
- Security Information and Event Management (SIEM) systems
 - about, 390
 - as real-time monitoring systems, 123
 - for incident response, 391
 - logging events fed into, 51
 - monitoring logs, 593
 - monitoring physical environment using, 53
- security models, computer systems
 - about, 337–338
 - Biba model, 340–342
 - BL model, 339
 - BN model, 343
 - CW model, 342–343
 - Graham-Denning model, 343
 - information flow model, 339
 - noninterference model, 339
 - state machine model, 338
 - Take-Grant model, 343
- security modes, MAC, 108–109
- security objectives and controls
 - about, 5–7
 - reducing risk of litigation, 12–13
 - understanding countermeasures and controls, 10–12
 - understanding risk modeling, 8–10
- security program for enterprise, 332
- security program, implementing
 - about, 34
 - assigning enforcement responsibilities, 44–45
 - assigning value to assets, 43
 - classifying data in, 38–41
 - components in, 34
 - defining category criteria, 41
 - defining classification categories, 40–41
 - defining required protective controls, 41–43
 - elemental phases in information life cycle, 37–38
 - implementing hiring practices, 45–47
 - implementing termination practices, 47–48
 - in risk assessment, 34
 - inventorying information assets, 43
 - managing third-party service providers, 50–51
 - monitoring and auditing in, 51–54
 - providing security awareness training program, 44, 49
 - reappraising and adjusting classification of information assets, 44
 - understanding organization chart, 36–37
- Security Reference Monitor (SRM), 110, 345–346
- Security Target, as component in evaluation and certification process, 348
- security team, approving provisioning process, 104
- security through obscurity control, 12
- security zones in facility, as physical control in designing physical security, 255–256
- seed, as nonsecret variable, 165
- segment data stream, 424
- SEI (Software Engineering Institute)
 - development of software capability maturity model integration, 587
 - introduced software IDEAL model, 588
- self-replicating exploit code, 629
- self-service password reset, 79
- SELinux operating system, MAC implementation in, 107
- SENSITIVE BUT UNCLASSIFIED clearance label, 612, 614
- sensors, network
 - in and outside external firewall, 128
 - in promiscuous mode, 128
 - mechanisms to defeat IDS, 127
- separation of duties
 - detering fraud using, 72–73
 - in fraud protection, 658–659
 - in software development security, 587
- separation of duties, in software development, 581
- Serial Line Interface Protocol (SLIP), 489
- server and client application architecture, 601–602
- server farm, 681
- Server Message Blocks (SMB), 680
- server redundancy, 681–682
- server vs. client, 428
- service accounts, 698
- Service-Level Agreements (SLAs), 275, 549, 683
- service mark, unregistered, 375
- Service-Oriented Architecture (SOA), 328–332
- Service Set Identifier (SSID), 90–91
- SESAME (Secure European System for Applications in a Multivendor Environment), 100–101
- session cookies, 637
- session hijacking, information theft by, 509
- Session Initiation Protocol (SIP), 491
- session keys, 164

Session layer of OSI Model

- Session layer of OSI Model, 423–424
- session tickets, KDC generating, 96–98
- SET (Secure Electronic Transaction), 233–234, 379
- SFTP (Secure FTP), 224, 233, 479
- shadow file, accessing, 698
- Shamir, Adi, 205
- shareware, 384
- Sherwood Applied Business Security Architecture (SABSA), 17, 334
- shielded twisted-pair (STP) cabling
 - about, 447
 - in designing physical security, 262
- shielding emanations, 443
- Shift (Caesar) cipher, 150–151
- ShiftRows function, 178
- shotgun attacks, 366, 693
- shoulder surfing, 81
- S-HTTP (Secure Hypertext Transfer Protocol), 232
- SIEM (Security Information and Event Management) systems
 - about, 390
 - for incident response, 391
 - logging events fed into, 51
 - monitoring logs, 593
 - monitoring physical environment using, 53
- Sigba machine, US, 154
- signage, in designing physical security, 266–267
- signal degradation, 444
- signature-based detection, 124, 637–638
- signature dynamics, authentication using, 85
- signing messages
 - using asymmetric key algorithms, 192–195, 198–201
 - using symmetric key algorithms, 185–189
- Simple Mail Transfer Protocol (SMTP)
 - about, 480
 - OSI Model Layer 7, 421
- Simple Network Management Protocol (SNMP)
 - about, 480
 - firewall rule and, 114–115
- Simple Object Access Protocol (SOAP), 603
- simplex, 423
- SIM (Subscriber Identity Module) card, attacks using, 513
- simulation testing, 567
- Single Loss Expectancy (SLE), 29–30
- single points of failure, identifying, 655, 673–674
- Single Sign On (SSO), 93–94, 602
- SIP (Session Initiation Protocol), 491
- SKEME (Secure Key Exchange Mechanism), 226
- slack space, clone disk, 403–404
- slash notation, 435
- SLAs (Service-Level Agreements), 275, 549, 683
- SLE (Single Loss Expectancy), 29–30
- SLIP (Serial Line Interface Protocol), 489
- smart cards, to authenticate users, 81–82
- SMB (Server Message Blocks), 680
- S/MIME (Secure Multipurpose Internet Message Extensions), 234, 480
- SM superscript for unregistered service mark, 375
- SMTP (Simple Mail Transfer Protocol)
 - about, 480
 - OSI Model Layer 7, 421
- Smurf attack, DoS, 506
- sniffing/eavesdropping, information theft by, 508
- SNMP (Simple Network Management Protocol)
 - about, 480
 - firewall rule and, 114–115
- SOAP (Simple Object Access Protocol), 603
- SOA (Service-Oriented Architecture), 328–332
- social engineering
 - as type of cybercrime, 368, 665–666
 - attack on passwords, 81, 145
- social networks, reviewing information of prospective employees from, 46
- sockets, 430
- soda acid, for fire suppression, 289–290
- Software as a Service (SaaS), 50, 331, 465
- software bugs
 - about, 582
 - life cycle of software with, 583
- Software Development Life Cycle (SDLC), 579–586
- software development security
 - about need for, 578–579
 - CASE tools, 590
 - five stages of CMMI, 587–588
 - logging requirements, 592–593
 - project-planning models, 588–590
 - separation of duties, 587
 - software development life cycle and, 579–586
 - software escrow, 593–594
 - software testing, 590–591
 - software updating, 591–592
- Software Engineering Institute (SEI)
 - development of software capability maturity model integration, 587
 - introduced software IDEAL model, 588

- software guard, access control, 119
- software licensing, managing, 383–384
- someplace you are, authentication category, authenticating by their proximity, 89
- something you are, authentication category, 84–89
- something you have, authentication category, 81–84
- something you know, authentication category, 77–81
- SONET (Synchronous Optical Network), 489
- SOX (Sarbanes-Oxley Act) of 2002, 17, 351, 372, 377
- span port, sensors attaching to layer-2 MAC switches using, 128
- Specified Area Border Routers (ABRs), 482
- speculation, managing, 33
- spiders, 635
- spike, electricity, 277
- spiral model, software project-planning models, 589
- split DNS, 474
- split-horizon route advertisements, 482
- split tunneling, 484
- spray and pray attacks, 693
- spread spectrum, 495
- sprinkler systems, types of, 284–286
- spurious noise, 278
- spyware, 630
- SQL injection attack, 625
- SRAM (Static Random Access Memory), 310, 312
- SRM (Security Reference Monitor), 110, 345–346
- SSH (Secure Shell), 223, 479, 483
- SSL (Secure Sockets Layer)
 - about, 7, 483
 - certificate, 230
 - data in transit protection over Internet, 141
 - vulnerability in, 696
 - web-based authentication and, 101
- SSO (Single Sign On), 93–94, 602
- SSTP (Secure Socket Tunneling Protocol), 229, 484
- Standard Ethernet networks, 446–448
- standard format, 422
- standardized documents, 16
- standard of behavior and activity of workers, 15–16
- standby UPS, 276
- state attack, 328
- stateful applications, 681–682
- stateful inspection, generation 3 firewall, 469
- stateless applications, 681–682
- state machine model, computer systems, 338
- static binding, in object-oriented programming, 598
- static electricity, 277
- static heuristics, 126
- Static Random Access Memory (SRAM), 310, 312
- static routing protocols, 481
- static separation of duties, deterring fraud using, 72, 659
- Statutes-at-Large, US, 373
- steganography, 234–235, 405
- stolen or lost authentication devices, users reporting, 84
- Storage Area Networks (SAN), 679–680
- Storage as a Service (SaaS), 679
- storage location, of backup data, 553
- storage media. *See also* hard disk reusing, 689–690
- storage phase, in information life cycle, 37
- STP (shielded twisted-pair) cabling
 - about, 447
 - in designing physical security, 262
- strategic goals, definition of, 5
- stream ciphers
 - symmetric key block ciphers vs., 180
 - symmetric keystream ciphers
 - about, 154
 - RC4 as, 157, 175
- streaming converted files, to hide data, 405
- streams technique, 700
- structured walkthrough testing, 567
- Stuxnet, launched by IDF, 368
- suballocation of blocks, 403
- SubBytes function, 178
- subject as process, 64
- subjects access objects statements, in trusted path, 64
- subjects statement, in trusted path, 64
- subnet mask, as part of IP network, 431–432
- subnetting process, 477
- subnet-to-subnet VPN connectivity, 224–225
- subordinate CA, 212
- subpoenas, 398
- Subscriber Identity Module (SIM) card, attacks using, 513
- subscription services, 545–546
- Substitution box (S-box)
 - about, 172
 - function, 173, 178
 - symmetric key block ciphers using, 175–176

substitution ciphers

- substitution ciphers
 - AES performing, 178
 - arbitrary, 151
 - Blowfish performing, 179
 - cryptanalysis and, 151
 - Enigma machine as, 154
 - hieroglyphics as, 149
 - IDEA performing, 175
 - symmetric key algorithms performing, 171
 - symmetric key block ciphers using, 175–176
 - Twofish performing, 179
 - Vigenere, 152–153
- substitution functions, 178
- substitution mapping, 151
- superscalar, CPU chips, 309
- supply systems
 - about, 9
 - in performing risk assessment, 29
 - recovery of, 547–548
- supply system threats, 247, 539
- support system, computer in cybercrime as, 368
- surge, electricity, 277
- surge suppressors, 278
- surveys, bias in, 28
- SVC (Switched Virtual Circuit), 490
- swap file, hard disk drive location, 322
- Switched Virtual Circuit (SVC), 490
- switches, 461–462, 475
- symmetric key, 158
- symmetric key algorithms
 - about, 169–170
 - sealing messages using, 185, 189
 - signing using, 185–189
 - weakness in, 189
 - XOR function in, 158, 172–173
- symmetric key authentications
 - Kerberos. *See* Kerberos authentication
- symmetric key block ciphers
 - about, 175–176
 - AES as, 157, 178–179
 - Blowfish as, 179
 - DEA as, 177
 - DES as, 177
 - double DES (2DES) as, 157, 177
 - Feistel Network and, 159
 - list of, 175–176
 - Lucifer algorithm, 156
 - modes of, 180–184
 - RC4 as, 175
 - RC5 as, 179
 - RC6 as, 179
 - S-box function, 172–173, 175–176, 178
 - stream ciphers vs., 180
 - triple DES (3DES or TDES) as, 177–178
 - Twofish as, 179
- symmetric key cryptography
 - about, 141
 - Kerberos using, 98
 - key in
 - about, 159
 - distribution of, 161
 - length of, 160–161
 - quantities of, 162
 - nonrepudiation and, 140
 - SESAME using, 100
 - using to delete files securely, 688
- symmetric key cryptosystem, signing and sending
 - services in, 171
- symmetric keystream ciphers
 - about, 154, 172–174
 - ARCFOUR (ARC4) as, 175
 - list of, 174–175
 - RC4 as, 157, 175
- symmetric key transposition ciphers, Scytale as, 150
- Synchronous Data Link Control (SDLC), 463
- Synchronous Optical Network (SONET), 489
- synchronous token devices, to authenticate users, 82–83
- synchronous transmissions, 453
- SYN packets, DoS attacks with, 506
- SYN (Synchronize), TCP segment, 426, 427–428
- sysklogd daemon, 134
- system authentication, based on symmetric keys, 185
- system cold start, 346
- system custodian role
 - in classifying data, 39
 - in IT, 651–652
- system design, in software development life cycle, 580
- system high-security mode, MAC model, 108, 325
- systems assets, vulnerabilities of, 8, 9
- system-specific policies, 15
- system takeover, 368

T

- T1 connections, 490
- T3 connections, 490
- tables, database
 - columns in, 605–606
 - data fields in, 605
 - rows in, 605
 - schema of, 606
- table-top testing, 567
- tab-separated value (.tsv) files, 610
- TACACS (Terminal Access Controller Access Control System), 117
- tactical goals, definition of, 5
- tailgating (piggybacking) method, 265
- Take-Grant model, computer system, 343
- tangible assets, in performing risk assessment, 24–25
- tape drives, recovering data using, 555
- tape vaulting, 558
- targeted attacks, anatomy of, 693–701
- targeted spot-checking users, 71
- target hardening method, in designing physical security
 - about, 257
 - CCTV cameras in, 267–270
 - emanations protection, 261–263
 - fences, 259–261
 - full wall vs. partition, 257
 - guard dogs as part of, 265
 - key management process, 259
 - lighting in, 267
 - locks, 258–259
 - physical access controls, 265–266
 - securing portable devices, 270–272
 - security guard considerations, 263–264
 - signage in, 266–267
 - window design, 257
- Target of Evaluation (ToE), as component in evaluation
 - and certification process, 348
- target selection, in targeted attacks, 694
- target system, computer in cybercrime as, 368
- Task Manager, Windows
 - displaying processes in, 320
 - identifying applications and processes with memory
 - leaks in, 320
- TCB (Trusted Computing Base), 345
- TCP/IP Protocol suite, 224, 441–442, 478–479
- TCP (Transmission Control Protocol)
 - about, 425–426, 478
 - vs. UDP, 425
- TCSEC (Trusted Computing System Evaluation Criteria), 345–347, 600
- TDD (Time-Division Duplexing), 504
- TDES (triple DES), 177–178
- TDMA (Time Division Multiple Access), 504
- TDM (Time-Division Multiplexing), 456–457, 491
- Teardrop attack, DoS, 506
- technical access controls
 - about, 11, 66
 - DAC as, 109–112
 - functional security objectives and, 70
 - MAC and, 105, 108
 - role-based access control as, 113
 - rule-based access control, 114–115
 - using to defend enterprise against fraud, 660
- technical detective controls, for fraud protection, 660
- technical report, in penetration testing, 669
- technical threats
 - about, 9, 248, 539
 - in performing risk assessment, 29
- telephone cramming, 514
- telephone slamming, 514
- Telnet, 223, 479, 694
- temperature and humidity considerations, in designing
 - physical security, 274–275
- Tempest, project codenamed, 444
- tempest technologies, 508
- temporal access controls, 119–120
- Temporal Key Integrity Protocol (TKIP), 498, 500
- Ten Commandments of Computer Ethics (CEI), 19
- Terminal Access Controller Access Control System (TACACS), 117
- termination practices, implementing, 47–48
- testing
 - acceptance, 581, 590
 - integration, 580, 590
 - penetration, 662–669
 - regression, 583, 591
 - software development life cycle installation
 - and, 580–581
 - unit, 567, 580, 590
 - validation, 567, 591
 - verification, 591
- Testing and Evaluation, as component in evaluation and
 - certification process, 348–349
- testing recovery plans, 567
- TFN (Tribe Flood Network), DDoS attack, 508
- TFTP (Trivial File Transfer Protocol), 479

TGS (Ticket Granting service)

- TGS (Ticket Granting service)
 - Kerberos support of, 94–95
 - Kerberos using, 96
- TGT (Ticket Granting Ticket)
 - as user access token, 96
 - weakness of Kerberos using, 98
- theft, separation of duties in avoiding, 72
- thermal fire detectors, 283
- thin clients (dumb terminals), 454, 463
- third-party cookies, 637
- third-party service providers
 - governance of, 382–383
 - managing, 50–51
- threads, computer, 309, 317
- threats
 - about, 8
 - categories of, 539
 - external, 9
 - internal, 9
- three-layer artificial neural network, 624
- three-way-handshake, TCP flags performing, 427–428
- thresholds, clipping level, 126
- Ticket Granting service (TGS)
 - Kerberos support of, 94–95
 - Kerberos using, 96
- Ticket Granting Ticket (TGT)
 - as user access token, 96
 - weakness of Kerberos using, 98
- tight cohesion, 599
- tight coupling, 599
- Time-Division Duplexing (TDD), 504
- Time Division Multiple Access (TDMA), 504
- time-division multiplexing, computer system, 309
- Time-Division Multiplexing (TDM), 456–457, 491
- Time of Check/Time of Use (TOC/TOU) attack, 628
- timing attacks, 627–628
- TKIP (Temporal Key Integrity Protocol), 498, 500
- TLS (Transport Layer Security), 101, 230, 483
- TM superscript for unregistered trademarks, 375
- TNI (Trusted Network Interpretation), 347
- TOC/TOU (Time of Check/Time of Use) attack, 628
- ToE (Target of Evaluation), as component in evaluation
 - and certification process, 348
- token devices, to authenticate users, 82–83
- token-passing bus method, 459
- token-passing ring method, 459
- token ring, 486–487
- TOP SECRET clearance label, 612–614
- tort law (civil law), 371, 372
- TPM (Trusted Platform Module) chip, 313
- tracking, in incident response system, 394
- tracks, hard disk, 403
- trademarks, 375
- trade secrets, 375
- traffic analysis, information theft by, 508, 631
- training, fire evacuation, 291
- training preventive controls and disaster recovery techniques and strategies, 568–569
- tranquility principle
 - about, 107
 - MAC and, 107
- transaction journaling, 554
- transaction processing, 614–617
- trans-border information flow, privacy protection
 - in, 380–381
- transferring risk, countermeasure of, 22–23, 32–33
- transient noise, 278
- transient noise, in signal transmission, 262
- transient viruses, 629
- Transmission Control Protocol (TCP)
 - about, 425–426, 478
 - three-way-handshake, 427–428
 - vs. UDP, 425
- transponder, 82
- transportation, in evidence life cycle, 397
- Transport layer of OSI Model, 424–430
- Transport Layer Security (TLS), 101, 230, 483
- Transport mode, IPsec in, 227
- transposition ciphers (permutation ciphers)
 - about, 145
 - AES performing, 178
 - Blowfish performing, 179
 - IDEA performing, 175
 - Scytale performing, 150
 - symmetric key algorithms performing, 171
 - Twofish performing, 179
- transposition functions, symmetric key block ciphers
 - using, 175–176
- Treadway, James C., Jr., 335
- trends and patterns, identifying, 621
- Tribe Flood Network (TFN), DDoS attack, 508
- triple DES (3DES or TDES), 157, 177–178
- Trivial File Transfer Protocol (TFTP), 479
- Trojan horse, 630–631
- Trojan horse executables, in targeted attacks, 699
- Trojan horse software, information theft using, 510

- Trojan, quarantining, 125
- TrueCrypt application, as encryption tool, 141
- Trusted Computing Base (TCB), 345
- Trusted Computing System Evaluation Criteria (TCSEC), 345, 600
- Trusted Network Interpretation (TNI), 347
- trusted path, 64–65
- Trusted Platform Module (TPM) chip, 313
- trusted recovery attempt, 346
- Trusted Root Certification Authorities Store, 214–215
- trust vs. privilege, 96
- truth tables, 158
 - about, 172
 - using in XOR function, 173
- .tsv (tab-separated value) files, 610
- tunneling protocols, 223
- Tunnel mode, IPsec in, 228
- tuple (records), 605
- twisted-pair cabling
 - about, 446–448
 - in designing physical security, 262
- two-factor (multi-factor) authentication, 89–90
- Twofish, as symmetric key block cipher, 179
- two-phase commit, 616
- Type I error (False rejection), in biometric system, 85
- Type II error (False acceptance), in biometric system, 85

U

- Ubuntu Linux, managing log file in, 134
- UDP (User Datagram Protocol)
 - about, 426–427, 478
 - vs. TCP, 425
- UMTS (Universal Mobile Telecommunications Systems), 504
- unauthorized system or data access, 628
- unauthorized users, logging system protections
 - against, 123
- uncertainty, managing, 33
- UNCLASSIFIED clearance label, 612, 614
- understanding risk modeling, 8
- Unicast mode, 459
- unintended (covert) communications channels, 627
- Uninterruptible Power Supply (UPS), 276, 548
- unipolar signaling, 452–453
- uni-processing systems, CPU chips, 309
- unit testing, in software development, 567, 580, 590
- Universal Mobile Telecommunications Systems (UMTS), 504
- Universal Serial Bus (USB) ports, 83, 312
- UNIX
 - DAC implemented in, 112
 - operating systems, 463
- unlawful search and seizure, 398
- unlicensed software, 384
- unregistered service mark, 375
- unregistered trademarks, 375
- Unshielded Twisted Pair (UTP) cabling
 - about, 446–447
 - in designing physical security, 262
- updates, of policy documents, 21
- update, software, 582, 591–592
- UPS (Uninterruptible Power Supply), 276, 548
- URG (Urgent), TCP segment, 426
- URL jumping, 328
- US Army, using cryptography, 154
- USB (flash memory) drives, 310, 689
- USB (Universal Serial Bus) ports, 83, 312
- US Code, 373
- US Congress, 373
- US Department of Defense Computer Security Center, 345
- US Department of Homeland Security, risk assessment
 - on DNS systems, 474
- use phase, in information life cycle, 38
- US Equal Opportunity Employment Commission (U.S. EEOC), on discrimination-related lawsuits, 15
- user accounts
 - locking down or disabling, 71
 - provisioning, 656–657
- User Datagram Protocol (UDP)
 - about, 426–427, 478
 - vs. TCP, 425
- user mode, operating system in, 314–315
- user provisioning life cycle, 656–657
- user role
 - in classifying data, 39
 - IT and, 652
- users
 - authenticating, 75. *See also* authentication categories
 - defining privileges of, 103
 - documenting provisioning process, 104
 - identity in DAC, 111
 - targeted spot-checking, 71
 - TGT as access token for, 96

US National Computer Security Center, 345
US privacy laws and regulations, 377
US Sigba machine, 154
US Statutes-at-Large, 373
UTP (Unshielded Twisted Pair) cabling
 about, 446–447
 in designing physical security, 262

V

vacations, deterring fraud using mandatory, 73
validation testing, in software development, 567, 591
Variable Length Subnet Masking (VLSM), 477
ventilation systems, producing positive air pressure, 275
verification, in evidence life cycle, 397
verification (one-to-one) method, in biometric system, 87–88
verification testing, in software development, 591
Vernam cipher (one-time pad), 154
Vernam, Gilbert, 172
VFR (Virtual Fragment Reassembly), 435
victim of computer crime, 368
videos, for monitoring environment, 51
Vigenere cipher, 152–153
Virtual Fragment Reassembly (VFR), 435
virtualization, is SOA, 329–330
Virtual Local Area Network (VLAN), 487
Virtual Machine (VM)
 honeypot implemented on, 129
 in SOA, 329–330
virtual memory, 311
Virtual Memory Manager (VMM), 321, 322–323
virtual password
 hash value (message digests) and, 78
 vs. passphrase, 78
Virtual Private Network (VPN)
 about, 475
 as secure channels for LAN-based applications, 223
 authentication accessing from, 90–91
 data in transit over, 7
 data in transit protection on, 141
 encrypted-channel, 69
 IPsec technology, 224–229
 L2TP technology, 228–229
 list of technologies, 223
 PPTP technology, 224
 protocols, 482–484
 SFTP technology, 224
 SSTP technology, 229
 using portable devices on, 270
 viruses, 629, 637
 VLAN (Virtual Local Area Network), 487
 VLSM (Variable Length Subnet Masking), 477
 VMM (Virtual Memory Manager), 321, 322–323
 VM (Virtual Machine)
 honeypot implemented on, 129
 in SOA, 329–330
 Voice over Internet Protocol (VoIP), 491–492
 voiceprint, authentication using, 85
 VoIP media gateway, 491
 VoIP (Voice over Internet Protocol), 491–492
 VPN (Virtual Private Network)
 about, 475
 as secure channels for LAN-based applications, 223
 authentication accessing from, 90–91
 data in transit over, 7
 data in transit protection on, 141
 encrypted-channel, 69
 IPsec technology, 224–229
 L2TP technology, 228–229
 list of technologies, 223
 PPTP technology, 224
 protocols, 482–484
 SFTP technology, 224
 SSTP technology, 229
 using portable devices on, 270
 vulnerabilities
 countermeasures for reducing or eliminating, 22–23, 31
 identification as starting position of attacker, 666
 of systems and information assets, 8, 9
 scanning for, 661–662
 vulnerability assessments
 about, 661
 incident response, 670
 penetration testing, 662–669
 privileged users assessment, 662
 vulnerability scanning, 661–662
 vulnerability databases, 696
 vulnerability scanning, as auditing functions, 124

W

- walkthroughs and inspections, performing periodic, 279–280
- walls, full vs. partition, in target hardening method of designing physical security, 257
- WAN (Wide Area Network)
 - about, 489–491
 - L2TP on, 229
 - privacy issues in trans-border information flow over, 380–381
- war chalking symbols, attacks on wireless networks using, 512
- war dialing, 695
- war driving, 695
- warm sites, leased, 545
- warrants, 398
- water detectors, in designing physical security, 279
- waterfall model, software project-planning models, 589
- water, for fire suppression, 289
- Wavelength-Division Multiplexing (WDM), 457
- web applications
 - about, 602
 - architecture in, 328
 - attacks on, 632–634
 - single sign on for, 602
- web-based authentication, 101–102
- web cache poisoning, 634–635
- web crawler, 635
- web of trust model, 222
- webpages, hijacking, 635
- web servers, 102
- WEPCrack password-cracking tool, 512
- WEP (Wired Equivalent Privacy), 499–500
- wet chemicals, for fire suppression, 290
- wet pipe sprinkler systems, 285
- white box testing, 667
- white-hat hackers, 581
- white noise, as emanations protection, 263, 443
- whole disk encryption, 313
- Wide Area Network (WAN)
 - about, 489–491
 - L2TP on, 229
 - privacy issues in trans-border information flow over, 380–381
- Wi-Fi networks. *See* wireless networks
- Wi-Fi Protected Access version 2 (WPA2), 500
- Wi-Fi Protected Access (WPA), 500
- WiMAX (Worldwide Interoperability for Microwave Access), 488–489, 503
- window design, in target hardening method of designing physical security, 257
- Windows
 - ACL implemented in, 110
 - authentication service running on, 75
 - DAC implemented in, 112
- Windows Internet Name Service (WINS), 422, 473
- Windows Server 2012, MAC/DAC hybrid operating system, 108
- Windows Task Manager
 - displaying processes in, 320
 - identifying applications and processes with memory leaks in, 320
- Windows XP, automatic teller machines running, 119
- WINS (Windows Internet Name Service), 422, 473
- wiping disk, securing portable devices by, 271–272
- Wired Equivalent Privacy (WEP), 499–500
- wireless access point, 462, 695
- wireless communications, transmitting data through, 262
- Wireless Fidelity Alliance, 494–495
- wireless LAN, 492–493
- wireless networks
 - 802.11i enterprise authentication, 501–503
 - about, 449, 487, 492–493
 - attacks on, 511–512
 - authentication accessing from, 90–91
 - basics of, 493–498
 - cellular networking vs., 504–505
 - frequency allocation for, 493
 - hazardous, 7
 - licensing ranges of WiMAX, 503
 - MIMO data streams in 802.11n and 802.11ac, 503
 - security for, 498–499
 - TKIP and, 498
 - using Diameter on, 118–119
 - vulnerabilities of, 499–500
 - Wi-Fi Protected Access version 2 and, 500
- Wisdom level, in knowledge pyramid, 622
- workers. *See* employees
- work factor, of cryptosystem, 144, 147–148
- workspace recovery, 543–544
- World Trade Organization (WTO), patents and, 374
- Worldwide Interoperability for Microwave Access (WiMAX), 488–489, 503
- World Wide Web (WWW), 458

worm

worm, 629–630
WPA2 (Wi-Fi Protected Access version 2), 500
WPA (Wi-Fi Protected Access), 500
WTO (World Trade Organization), patents and, 374
WWW (World Wide Web), 458

X

X.25 connections, 490
X.500 Directory Services, 99–100
X.509 digital certificate, 100–101, 215–220
XML (Extensible Markup Language), 102, 603, 612
XOR (Exclusive Or) function, 158, 172–173, 175–176, 178
XSS (cross-site scripting) attacks, 632–634
XTACACS (eXtended TACACS), 117
XTR, 207

Z

Zachman Framework, 17
Zachman Framework, for enterprise architecture, 334
zero-day exploits, 629, 637, 696–697
zero knowledge proof, in mutual authentication, 91–93
Zigby, 486
Zimmermann, Phil, 222