Chapter 19 – Security

Outline

Chapter 19 – Security

Outline (continued)

Chapter 19 – Security

Outline (continued)

Objectives

- After reading this chapter, you should understand:
  - the role of authentication in providing secure systems.
  - access control models, policies and mechanisms.
  - public-key/private-key cryptography.
  - security and authentication protocols, such as SSL and Kerberos.
  - digital signatures, digital certificates and certificate authorities.
  - security threats, such as viruses, worms, exploits and denial-of-service attacks.
  - Virtual Private Networks and IPSec.

19.1 Introduction

- Computer security
  - Addresses the issue of preventing unauthorized access to resources and information maintained by computers
  - Encompasses the following issues:
    - Guaranteeing the privacy and integrity of sensitive data
    - Restricting the use of computer resources
    - Providing resilience against malicious attempts to incapacitate the system
  - Employs mechanisms that shield resources such as hardware and operating system services from attack

19.2 Cryptography

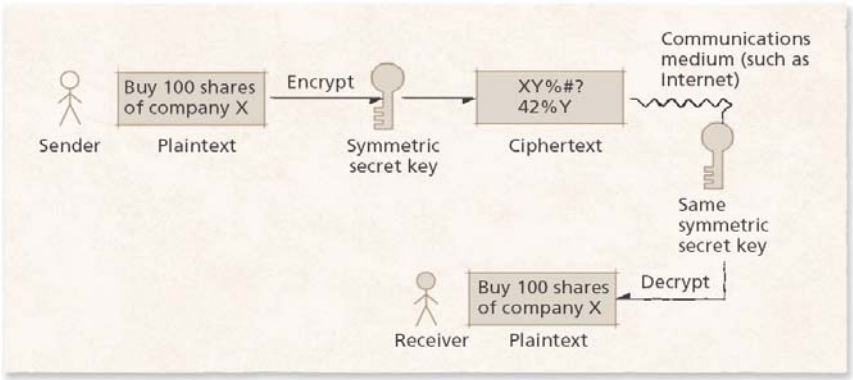
- Cryptography
  - Encoding and decoding data so that it can be interpreted only by the intended recipients
  - Data is transformed by means of a cipher or cryptosystem
    - Modern cryptosystems rely on algorithms that operate on the individual bits or blocks (a group of bits) of data, rather than letters of the alphabet
  - Encryption and decryption keys
    - Binary strings of a given length

19.2.1 Secret-Key Cryptography

- Secret-key cryptography
  - Also known as symmetric cryptography
  - Uses the same secret key to encrypt and decrypt a message
    - Sender
      - Encrypts a message using the secret key
      - Sends encrypted message to the intended recipient
    - Recipient
      - Decrypts the message using the same secret key

19.2.1 Secret-Key Cryptography

Figure 19.1 Encrypting and decrypting a message using a secret key.

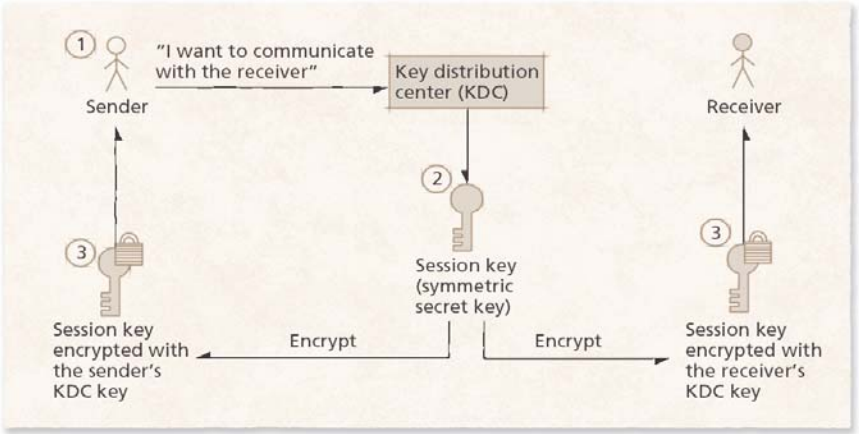


19.2.1 Secret-Key Cryptography

- Limitation of secret-key cryptography
  - Before two parties can communicate securely, they must find a secure way to exchange the secret key
    - Can be done by courier or a key distribution center (KDC)
      - KDCs generate session keys to clients
- Examples of secret-key cryptography:
  - DES
  - 3DES
  - AES

19.2.1 Secret-Key Cryptography

Figure 19.2 Distributing a session key with a key distribution center.

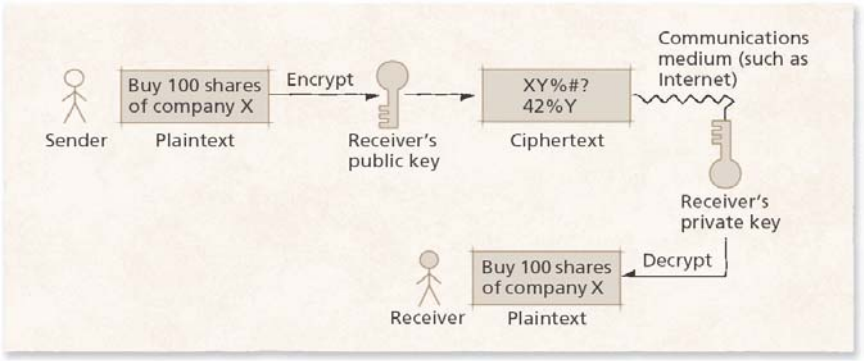


19.2.2 Public-Key Cryptography

- Public-key cryptography
  - Solves the problem of securely exchanging symmetric keys
  - Asymmetric
    - Employs two inversely related keys:
      - Public key
        - Freely distributed
      - Private key
        - Kept secret by its owner
  - If the public key encrypts a message, only the corresponding private key can decrypt it

19.2.2 Public-Key Cryptography

Figure 19.3 Encrypting and decrypting a message using public-key cryptography.

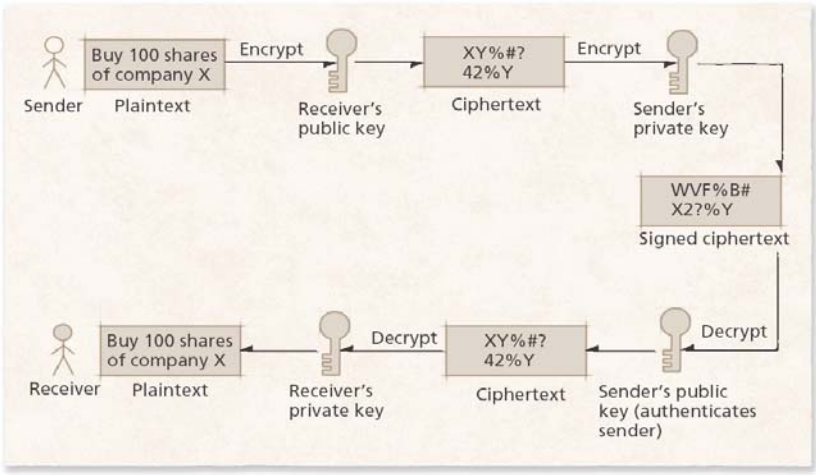


19.2.2 Public-Key Cryptography

- If the decryption key is the sender's public key and the encryption key is the sender's private key, the sender of the message can be authenticated
  - Message should be encrypted first using the receiver's public key, then with the sender's secret key
    - Public key provides confidentiality
    - Secret key provides authentication
- Examples of public-key cryptography:
  - RSA
  - Pretty Good Privacy (PGP)

19.2.2 Public-Key Cryptography

Figure 19.4 Authentication with a public-key algorithm.



19.3 Authentication

- Authentication
  - Identifying users and the actions they are allowed to perform
  - A user can be identified by:
    - a unique characteristic of the person (e.g., fingerprints, voiceprints, retina scans and signatures)
    - ownership of an item (e.g., badges, identification cards, keys and smart cards)
    - user knowledge (e.g., passwords, personal identification numbers (PINs) and lock combinations)

19.3.1 Basic Authentication

- Simple password protection
  - Most common authentication scheme
  - The user chooses a password, memorizes it and presents it to the system to gain admission to a resource or system
- Weaknesses of password protection
  - Users tend to choose passwords that are easy to remember
    - For example: the name of a spouse or pet
  - Someone who has obtained personal information about the user might try to log in several times using passwords that are characteristic of the user
    - Several repeated attempts might result in a security breach
- Password salting
  - Technique that inserts characters at various positions in the password before encryption
  - Can thwart attempts at recovering passwords from password files

19.3.1 Basic Authentication

Figure 19.5 Salting passwords (Base 64 encoding).

Plaintext	*Ciphertext*
password	cGFzc3dvcmQ=
psasaswlortd	cHNhc2Fzd2xvcnRk
newpassword	bmV3cGFzc3dvcmQ=
nsewaplatssewodrd	bnNld2FwbGF0c3Nld29kcmQ=

19.3.2 Biometrics and Smart Cards

- Biometrics
  - Uses unique personal information to identify a user
    - Fingerprints
    - Eyeball iris scans
    - Face scans
- Smart cards
  - Often designed to resemble a credit card
  - Can serve many different functions, from authentication to data storage
  - Most popular: memory cards and microprocessor cards

19.3.3 Kerberos

- Kerberos
  - Freely available, open-source protocol developed at MIT
  - Can provide protection against internal security attacks
  - Employs secret-key cryptography
    - To authenticate users in a network
    - To maintain the integrity and privacy of network communications

19.3.3 Kerberos

- Kerberos implementation
  - Uses an authentication server and a Ticket Granting Service to control access to network resources
    1. Client submits username/password to authentication server
    2. If valid, the authentication server issues a Ticket-Granting Ticket (TGT) encrypted with the client's secret key
    3. Client sends decrypted TGT to the TGS when requesting a resource. If valid, TGS issues a service ticket encrypted with client's secret key.
    4. Client decrypts service ticket, which it uses to access network resources

19.3.4 Single Sign-On

- Single sign-on
  - Simplifies the authentication process
    - Allows the user to log in once using a single password to access multiple applications across multiple computers
  - Important to secure single sign-on passwords
    - If a password becomes available to crackers, all applications protected by that password can be accessed and attacked
  - Available in forms of workstation login scripts, authentication server scripts and token-based authentication

19.4 Access Control

- Today's operating systems must
  - Carefully guard against unintentional and malicious use of computer resources
  - Protect operating system services and sensitive information from users and/or software that have gained access to computer resources
- Access rights
  - Protect system resources and services from potentially dangerous users
  - Restrict or limit the actions that can be performed on resources
  - Typically managed by access control lists or capability lists

19.4.1 Access Rights and Protection Domains

- The key to operating system security is to control access to system resources
- The most common access rights:
  - Read
  - Write
  - Execute
- Techniques employed to manage access rights:
  - Access control matrices
  - Access control lists
  - Capability lists

19.4.2 Access Control Models and Policies

- Security model
  - Defines a system's subjects, objects and privileges
  - Examples:
    - User classes
    - Role-based access control
  - Discretionary Access Control (DAC)
    - File owner controls permissions
  - Mandatory Access Control (MAC)
    - Predefine a central permission scheme

19.4.2 Access Control Models and Policies

- Security policy
  - Typically specified by the user and/or system administrator
  - Defines which privileges to objects are assigned to subjects
  - Most incorporate the principle of least privilege
    - Subject is granted access only to the objects it requires to perform its tasks
- Security mechanism
  - The method by which the system implements the security policy

19.4.3 Access Control Mechanisms

- Access control matrices
  - Match subjects and objects to the appropriate access rights
  - Concept behind the model is simple
  - Most systems contain many subjects and objects, resulting in a large matrix that is an inefficient means for access control

19.4.3 Access Control Mechanisms

Figure 19.6 Access control matrix for a small group of subjects and objects.

	File A	File B	Printer
Alice	Read* Write*	Read* Write*	Print*
Bob	Read* Write	Read* Write	Print
Chris	Read		Print
David		Read	
Guest			

19.4.3 Access Control Mechanisms

- Access control lists and capability lists
  - Derived from the principle of least privilege
  - Often more efficient and flexible methods of managing access rights
  - Capability
    - Pointer or token that grants access to a subject that possesses it

19.4.3 Access Control Mechanisms

Figure 19.7 Access control list derived from the access control matrix.

```
File A:
   <Alice, {read*, write*}>
   <Bob, {read*, write}>
   <Chris, {read}>
File B:
   <Alice, {read*, write*}>
   <Bob, {read*, write}>
   <David, {read}>
Printer:
   <Alice, {print*}>
   <Bob, {print}>
   <Chris, {print}>
```


19.5 Security Attacks

- Cryptanalytic attacks
- Viruses and worms
- Denial-of-service attacks
  - Domain name system (DNS) attack
- Software exploitation
  - Buffer overflow
- System penetration
  - Web defacing

19.5.1 Cryptanalysis

- Cryptanalytic attacks
  - Attempt to decrypt ciphertext without possessing the decryption key
  - Most common attack
    - Encryption algorithm is analyzed to find relations between bits of the encryption key and bits of the ciphertext
    - Goal is to determine the key from the ciphertext
  - Weak statistical trends between ciphertext and keys can be exploited to gain knowledge about the key
- Covertly recovered key can be used to decrypt every message that uses the key

19.5.2 Viruses and Worms

- Virus
  - Executable code often sent as an attachment to an e-mail message or hidden in files such as audio clips, video clips and games
  - Attaches to or overwrites other files to replicate itself
  - Can corrupt files, control applications or even erase a hard drive
  - Can be spread across a network simply by sharing “infected” files embedded in e-mail attachments, documents or programs
- Worm
  - Executable code that spreads by infecting files over a network
  - Rarely requires any user action to propagate
  - Does not need to be attached to another program or file to spread
- Once a virus or worm is released, it can spread rapidly, often infecting millions of computers worldwide within minutes or hours

19.5.3 Denial-of-Service (DoS) Attacks

- DoS attack
  - Prevent a system from servicing legitimate requests
  - In many DoS attacks, unauthorized traffic saturates a network's resources, restricting access for legitimate users
  - Typically, attack is performed by flooding servers with data packets
  - Usually require a network of computers to work simultaneously, although some skillful attacks can be achieved with a single machine
  - Can cause networked computers to crash or disconnect, disrupting service on a Web site or even disabling critical systems such as telecommunications or flight-control centers

19.5.4 Software Exploitation

- Buffer overflow attacks
  - Occurs when an application sends more data to a buffer than it can hold
  - Can push the additional data into adjacent buffers, corrupting or overwriting existing data
  - A well-designed buffer overflow attack can replace executable code in an application's stack to alter its behavior
  - May contain malicious code that will then be able to execute with the same access rights as the application it attacked
  - Depending on the user and application, the attacker may gain access to the entire system

19.5.5 System Penetration

- System penetration
  - Successful breach of computer security by an unauthorized external user
  - Always potentially dangerous, although a quick response can usually thwart an intruder's attack before any significant damage is done
  - Often occurs as a result of a Trojan horse, back-door program or an exploited bug in software or the operating system
  - Example
    - Web defacing
      - Crackers illegally obtain access to modify an organization's Web site and change the contents

19.6 Attack Prevention and Security Solutions

- Firewalls
- Intrusion detection systems
- Antivirus software
- Security patches
- Secure file systems

19.6.1 Firewalls

- Firewalls
  - Protect a local area network (LAN) from intruders outside the network
  - Police inbound and outbound traffic for the LAN
- Types of firewalls
  - Packet-filtering firewall
    - Inspects packets for inconsistencies such as incorrect source address
  - Application-level gateways
    - Inspect packets for malicious payloads

19.6.2 Intrusion-Detection Systems (IDSs)

- IDSs
  - Monitor networks and application log files
    - Logs record information about system behavior, such as:
      - The time at which operating system services are requested
      - The name of the process that requests them
  - Examine log files to alert system administrators of suspicious application and/or system behavior
  - If an application exhibits erratic or malicious behavior, an IDS can halt the execution of that process
- Host-based intrusion detection
- Network-based intrusion detection

19.6.3 Antivirus Software

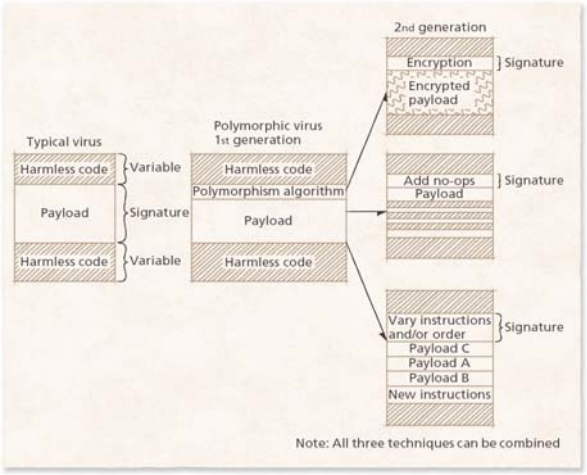
- Antivirus software
  - Attempts to protect a computer from a virus and/or identify and remove viruses on that computer
  - Various techniques used to detect and remove viruses from a system
    - None can offer complete protection

19.6.3 Antivirus Software

- Signature-scanning virus detection
  - Relies on knowledge about the structure of the computer virus's code
  - Uses a known virus list
    - Can be particularly ineffective against variants and polymorphic viruses
- Heuristic scanning
  - Can prevent the spread of viruses by detecting and suspending any program exhibiting virus-like behavior:
    - Replication, residence in memory and/or destructive code
  - Primary strength: it can detect viruses that have not yet been identified

19.6.3 Antivirus Software

Figure 19.8 Polymorphic virus.



19.6.4 Security Patches

- Security patches
  - Code releases that address security flaws
  - Simply releasing a patch for a security flaw is insufficient to improve security
  - Developers should address security flaws by:
    - Notifying their users quickly
    - Providing software that facilitates the process of applying security patches
      - Example: Hotfixes
        - Microsoft Automatic Updates

19.6.5 Secure File Systems

- Secure file systems
  - Protect sensitive data regardless of how the data is accessed
- Encrypting File System (EFS)
  - Uses cryptography to protect files and folders in an NTFS file system
  - Uses secret-key and public-key encryption to secure files

19.6.6 Orange Book Security

- The Orange Book
  - Officially named "Department of Defense Trusted Computer System Evaluation Criteria"
  - Designed to evaluate the security features of operating systems
  - Define levels of security in operating systems
  - Classifies systems into four levels of security protection
    - A, B, C and D
    - The lowest level of security is D and the highest is A

19.7 Secure Communication

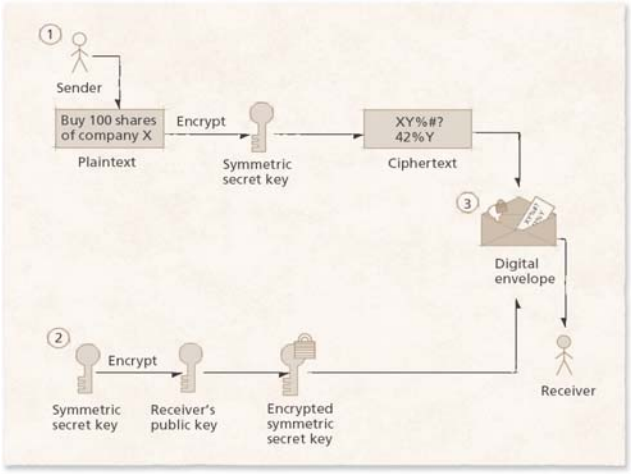
- Five fundamental requirements for a successful, secure transaction
  - Privacy
    - Ensuring that the information transmitted over the Internet has not been viewed by a third party
  - Integrity
    - Ensuring that the information sent or received has not been altered
  - Authentication
    - Verifying the identities of the sender and receiver
  - Authorization
    - Managing access to protected resources on the basis of user credentials
  - Nonrepudiation
    - Ensuring that the network will operate continuously

19.8 Key Agreement Protocols

- Public-key algorithms
  - Most often employed to exchange secret keys securely
- Key agreement protocol
  - The process by which two parties can exchange keys over an unsecure medium
  - Digital envelopes
  - Digital signatures (using the SHA-1 and MD5 hash algorithms)

19.8 Key Agreement Protocols

Figure 19.9 Creating a digital envelope.



19.8.1 Key Management

- Maintaining the secrecy of private keys is essential to the maintenance of cryptographic system security
- Most security breaches result from poor key management rather than cryptanalytic attacks
  - For example: The mishandling of private keys, resulting in key theft

19.8.1 Key Management

- Key generation
  - The process by which keys are created
  - Important to use a key-generation program that can generate a large number of keys as randomly as possible
  - Key security is improved when key length is large enough that brute-force cracking is computationally infeasible

19.8.2 Digital Signatures

- The electronic equivalents of written signatures
- Developed to address the absence of authentication and integrity in public-key cryptography
- Authenticate senders' identities
- Difficult to forge
- Hash value uniquely identifies a message
  - Examples
    - Secure Hash Algorithm (SHA-1)
    - MD5 Message Digest Algorithm
    - Digital Signature Algorithm (DSA)

19.9 Public-Key Infrastructure Certificates and Certificate Authorities

- Limitation of public-key cryptography
  - Multiple users might share the same set of keys, making it difficult to establish each party's identity
- Public Key Infrastructure (PKI)
  - Provides a solution by integrating public-key cryptography with digital certificates and certificate authorities to authenticate parties in a transaction
- Digital certificate
  - Digital document that identifies a user and is issued by a certificate authority (CA)

19.10 Secure Communication Protocols

- Developed to provide security in several layers of the traditional TCP/IP stack
- Secure Sockets Layer (SSL)
- Internet Protocol Security (IPSec)

19.10.1 Secure Sockets Layer (SSL)

- Nonproprietary protocol that secures communication between two computers on the Internet
- Implements public-key cryptography using the RSA algorithm and digital certificates
  - To authenticate the server in a transaction
  - To protect private information as it passes over the Internet
- SSL transactions do not require client authentication
  - Many servers consider a valid credit card number to be sufficient for authentication in secure purchases

19.10.2 Virtual Private Networks (VPNs) and IP Security (IPSec)

- Virtual Private Networks (VPNs)
  - Provide secure communications over public connections
  - Encryption enables VPNs to provide the same services and security as private networks
  - Created by establishing a secure communication channel over the Internet
- IPSec (Internet Protocol Security)
  - Uses public-key and symmetric-key cryptography
    - To ensure data integrity, authentication and confidentiality
  - Commonly used to implement a secure tunnel

19.10.3 Wireless Security

- Wireless devices
  - Limited bandwidth and processing power, high latency and unstable connections
  - Establishing secure wireless communication can be challenging
- Wired Equivalent Privacy (WEP) protocol
  - Protects wireless communication by encrypting transmitted data and preventing unauthorized access to the wireless network
  - Several drawbacks make it too weak for many environments
- Wi-Fi Protected Access (WPA)
  - Provides improved data encryption and enables user authentication, a feature not supported by WEP

19.11 Steganography

- Steganography
  - The practice of hiding information within other information
  - Can be used to hide a piece of information
    - For example: a message or image, within another image, message or other form of multimedia
- Digital watermarks
  - Used to protect intellectual property
  - Exploit unused portions of files to store hidden messages, while the digital files maintain their intended semantics

19.12 Proprietary and Open-Source Security

- Advantages of open-source security applications
  - Interoperability
    - Open-source applications tend to implement standards and protocols that many developers include in their products.
  - An application's source code is available for extensive testing and debugging by the community at large
- Weaknesses of proprietary security
  - Nondisclosure
  - The number of collaborative users that can search for security flaws and contribute to the overall security of the application is limited
- Proprietary systems, however, can be equally as secure as open-source systems

19.13 Case Study: UNIX Security

- UNIX security
  - Encrypted password file
    - When a user enters a password, it is encrypted and compared to the password file
    - Passwords are unrecoverable even by the system administrator
  - UNIX *setuid* permission feature
    - Program is run with the privileges of the owner of the file, who may not be the user executing the files
    - This powerful feature has security flaws
      - Particularly when the owner is has "superuser" privileges (access to all files in a UNIX system)