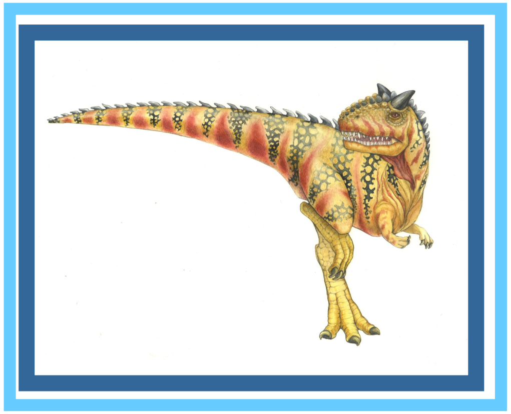
Chapter 14: Security



Chapter 14: Security

- The Security Problem
- Program Threats
- System and Network Threats
- Cryptography as a Security Tool
- User Authentication
- Implementing Security Defenses
- Firewalling to Protect Systems and Networks
- Computer-Security Classifications
- An Example: Windows XP

Objectives

- To discuss security threats and attacks
- To explain the fundamentals of encryption, authentication, and hashing
- To examine the uses of cryptography in computing
- To describe the various countermeasures to security attacks

The Security Problem

- Security must consider external environment of the system, and secure the system resources from misuse
- Intruders (crackers) attempt to breach security
- Threat is potential security violation
- **Attack** is attempt to breach security
- Attack can be accidental or malicious
- Easier to protect against accidental than malicious misuse

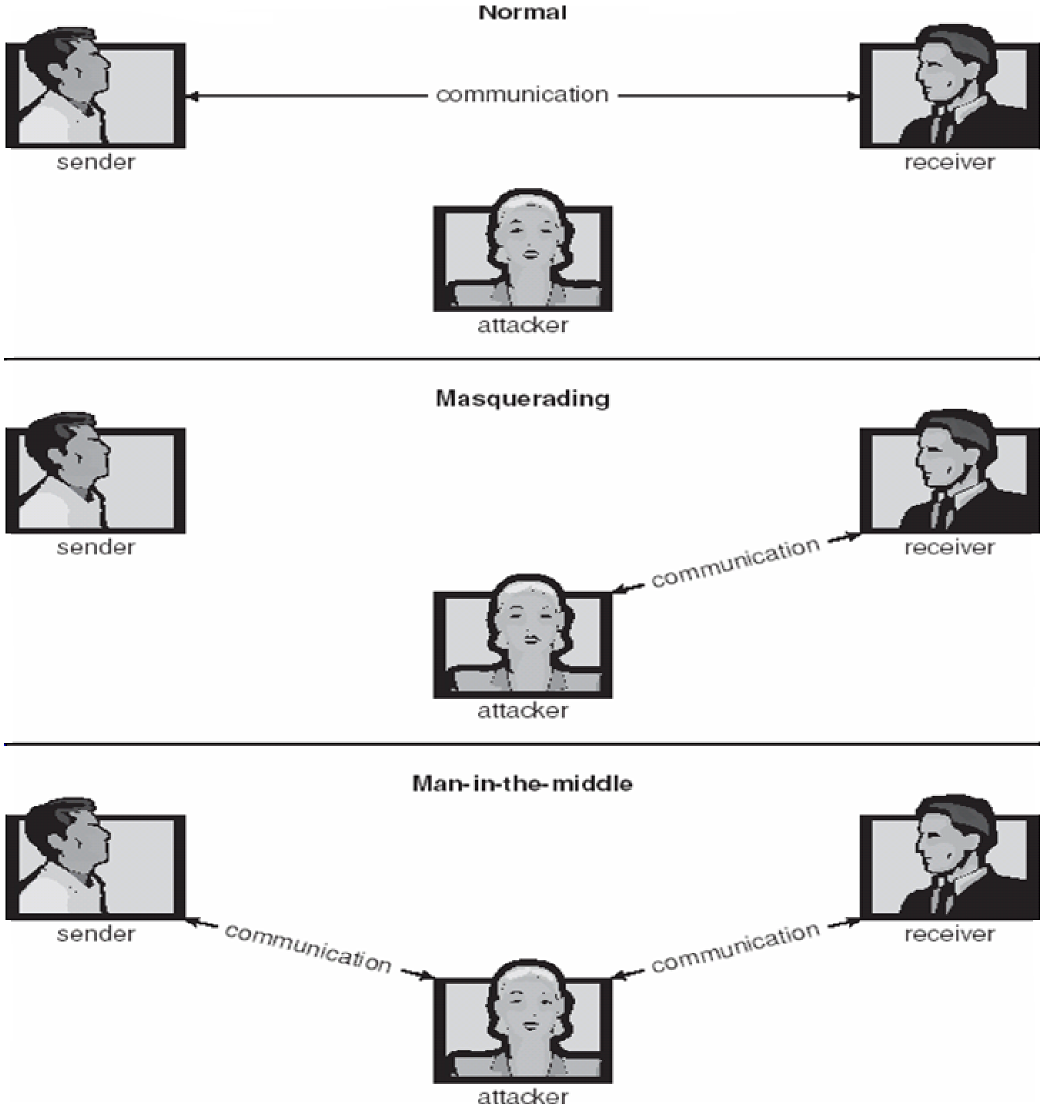
Security Requirements

- Commonly used attributes required in secure network transactions
  - **Authentication**
  - **Confidentiality**
  - **Integrity**
  - **Non-repudiation**

Security Violations

- Categories
  - **Breach of confidentiality**
  - **Breach of integrity**
  - **Breach of availability**
  - **Theft of service**
  - **Denial of service**
- Methods
  - **Masquerading (breach of authentication)**
  - **Replay attack**
    - **Message modification**
  - **Man-in-the-middle attack**
  - **Session hijacking**

Standard Security Attacks



Security Measure Levels

- Security must occur at four levels to be effective:
  - **Physical**
  - **Human**
    - Avoid social engineering, phishing, dumpster diving
  - **Operating System**
  - **Network**
- Security is as weak as the weakest link in the chain

Security Measure Levels

Social engineering,

in the context of security, is understood to mean the art of manipulating people into performing actions or divulging confidential information.

This is a type of confidence trick for the purpose of information gathering, fraud, or computer system access.

Security Measure Levels

Phishing

is the act of attempting to acquire information such as usernames, passwords, and credit card details (and sometimes, indirectly, money) by masquerading as a trustworthy entity in an electronic communication.

Communications purporting to originate from popular social web sites, auction sites, online payment processors or IT administrators are commonly used to lure the unsuspecting public.

Security Measure Levels

Dumpster diving (American English)

(skipping in British English) is the practice of sifting through commercial or residential waste to find items that have been discarded by their owners, but that may prove useful to the dumpster diver.

Program Threats

- **Trojan Horse**
  - Code segment that misuses its environment
  - Exploits mechanisms for allowing programs written by users to be executed by other users
  - Spyware, pop-up browser windows, covert channels
- **Trap Door**
  - Specific user identifier or password that circumvents normal security procedures
  - Could be included in a compiler
- **Logic Bomb**
  - Program that initiates a security incident under certain circumstances
- **Stack** and **Buffer Overflow**
  - Exploits a bug in a program (overflow either the stack or memory buffers)

Program Threats

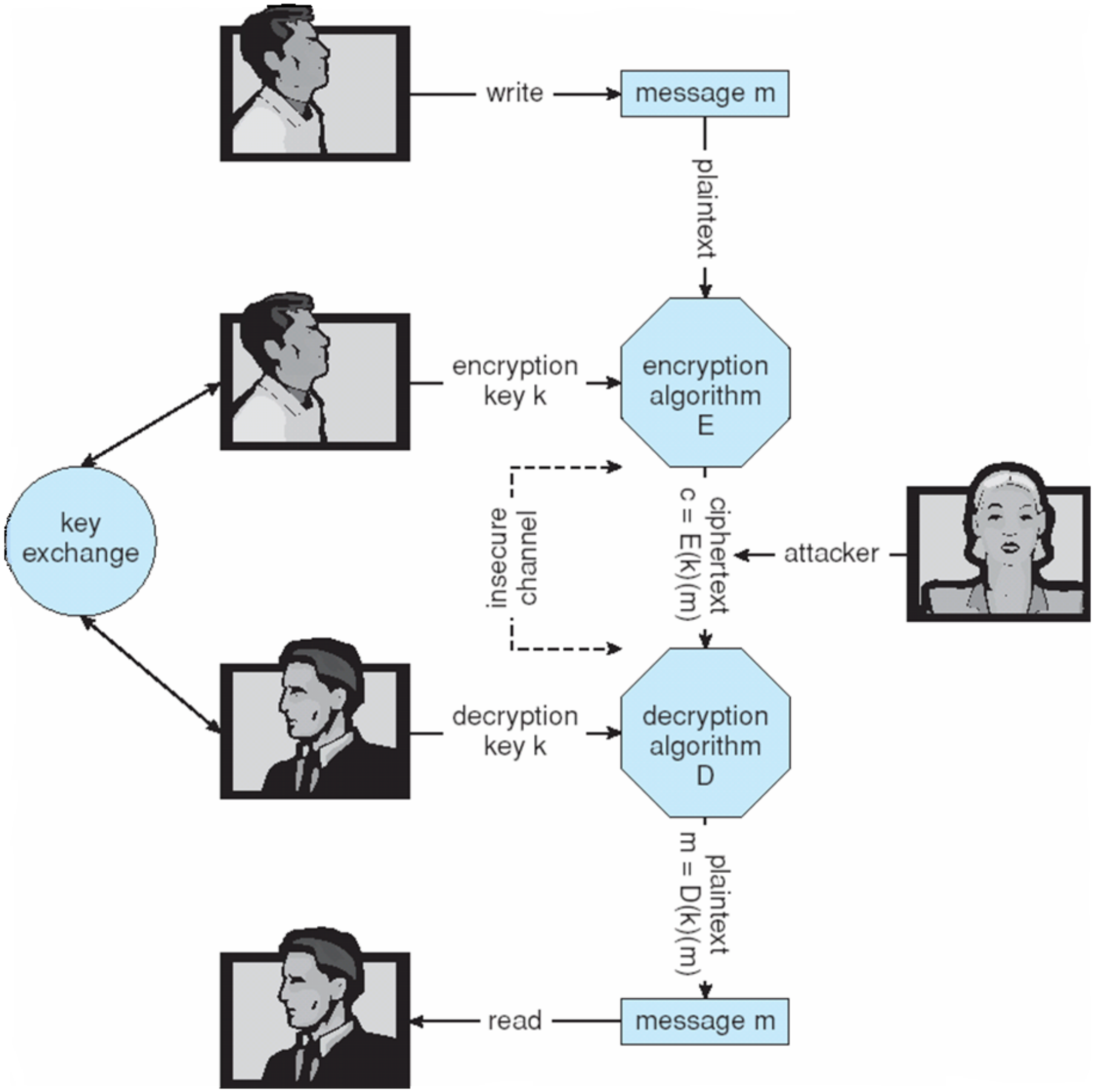
Covert channel

is a type of computer security attack that creates a capability to transfer information between processes that are not allowed to communicate by the computer security policy.

Cryptography as a Security Tool

- Broadest security tool available
  - Source and destination of messages cannot be trusted without cryptography
  - Means to constrain potential senders (*sources*) and / or receivers (*destinations*) of *messages*
- Based on secrets (keys)

Secure Communication over Insecure Medium



Encryption

- Encryption algorithm consists of
  - Set of K keys
  - Set of M Messages
  - Set of C ciphertexts (encrypted messages)
  - A function $E : K \rightarrow (M \rightarrow C)$. That is, for each $k \in K$, $E(k)$ is a function for generating ciphertexts from messages
    - Both E and $E(k)$ for any k should be efficiently computable functions
  - A function $D : K \rightarrow (C \rightarrow M)$. That is, for each $k \in K$, $D(k)$ is a function for generating messages from ciphertexts
    - Both D and $D(k)$ for any k should be efficiently computable functions
- An encryption algorithm must provide this essential property: Given a ciphertext $c \in C$, a computer can compute m such that $E(k)(m) = c$ only if it possesses $D(k)$.
  - Thus, a computer holding $D(k)$ can decrypt ciphertexts to the plaintexts used to produce them, but a computer not holding $D(k)$ cannot decrypt ciphertexts
  - Since ciphertexts are generally exposed (for example, sent on the network), it is important that it be infeasible to derive $D(k)$ from the ciphertexts

Asymmetric Encryption

- Public-key encryption based on each user having two keys:
  - public key – published key used to encrypt data
  - private key – key known only to individual user used to decrypt data
- Must be an encryption scheme that can be made public without making it easy to figure out the decryption scheme
  - Most common is RSA block cipher
  - Efficient algorithm for testing whether or not a number is prime
  - No efficient algorithm is know for finding the prime factors of a number

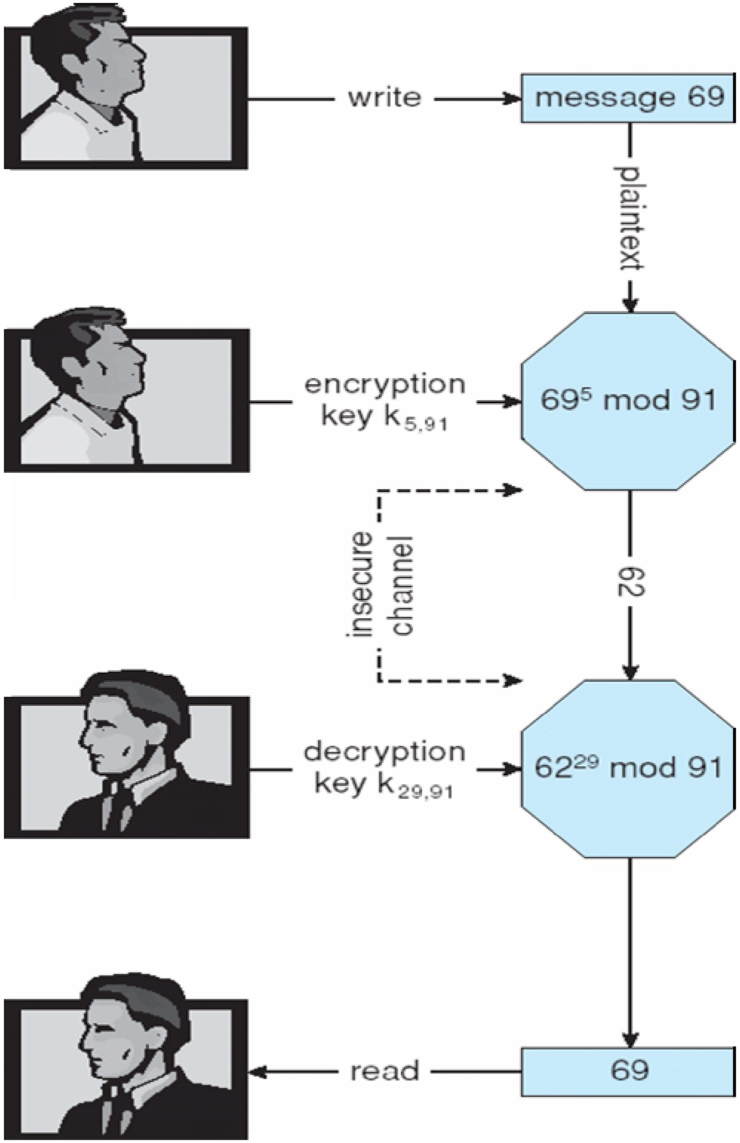
Asymmetric Encryption (Cont.)

- Formally, it is computationally infeasible to derive $D(k_d , N)$ from $E(k_e , N)$, and so $E(k_e , N)$ need not be kept secret and can be widely disseminated
  - $E(k_e , N)$ (or just k_e) is the public key
  - $D(k_d , N)$ (or just k_d) is the private key
  - N is the product of two large, randomly chosen prime numbers p and q (for example, p and q are 512 bits each)
  - k_e k_d are computed with extended Euclid algorithm (D. Knuth)
  - k_e satisfies $\text{GCD}(k_e , (p-1)(q-1)) = 1$
  - k_d satisfies $k_e k_d \bmod (p-1)(q-1) = 1$
  - Encryption algorithm is $E(k_e , N)(m) = m^{k_e} \bmod N$,
  - The decryption algorithm is then $D(k_d , N)(c) = c^{k_d} \bmod N$

Asymmetric Encryption Example

- For example, make p = 7and q = 13
- We then calculate $N = 7*13 = 91$ and $\varphi = (p - 1)(q - 1) = N - p - q + 1 = 72$
- We next select k_e relatively prime to 72 and< 72, yielding 5
- Finally,we calculate k_d such that $k_e k_d$ mod 72 = 1, yielding 29
- We how have our keys
  - Public key, $k_{e,}$ N = 5, 91
  - Private key, k_d , N = 29, 91
- Encrypting the message 69 with the public key results in the cyphertext 62
- Cyphertext 62 can be decoded with the private key yielding 69.
  - Public key can be distributed in cleartext to anyone who wants to communicate with holder of public key

Encryption and Decryption using RSA Asymmetric Cryptography



Authentication

- Constraining set of potential senders of a message
  - Complementary and sometimes redundant to encryption
  - Also can prove message unmodified
- Algorithm components
  - A set K of keys
  - A set M of messages
  - A set A of authenticators
  - A function $S : K \rightarrow (M \rightarrow A)$
    - That is, for each $k \in K$, $S(k)$ is a function for generating authenticators from messages
    - Both S and $S(k)$ for any k should be efficiently computable functions
  - A function $V : K \rightarrow (M \times A \rightarrow \{\text{true, false}\})$. That is, for each $k \in K$, $V(k)$ is a function for verifying authenticators on messages
    - Both V and $V(k)$ for any k should be efficiently computable functions

Authentication (Cont.)

- For a message m, a computer can generate an authenticator $a \in A$ such that $V(k)(m, a) = \texttt{true}$ only if it possesses $S(k)$
- Thus, computer holding $S(k)$ can generate authenticators on messages so that any other computer possessing $V(k)$ can verify them
- Computer not holding $S(k)$ cannot generate authenticators on messages that can be verified using $V(k)$
- Since authenticators are generally exposed (for example, they are sent on the network with the messages themselves), it must not be feasible to derive $S(k)$ from the authenticators

Authentication – Digital Signature

- Based on asymmetric keys and digital signature algorithm
- Authenticators produced are digital signatures
- In a digital-signature algorithm, computationally infeasible to derive $S(k_s)$ from $V(k_v)$
  - V is a one-way function
  - Thus, k_v is the public key and k_s is the private key
- Consider the RSA digital-signature algorithm
  - Similar to the RSA encryption algorithm, but the key use is reversed
  - Digital signature of message $S(k_s)(m) = H(m)^{k_s} \bmod N$
  - The key k_s again is a pair d, N, where N is the product of two large, randomly chosen prime numbers p and q
  - Verification algorithm is $V(k_v)(m, a) \equiv (a^{k_v} \bmod N = H(m))$
    - Where k_v satisfies $k_v k_s \bmod (p - 1)(q - 1) = 1$

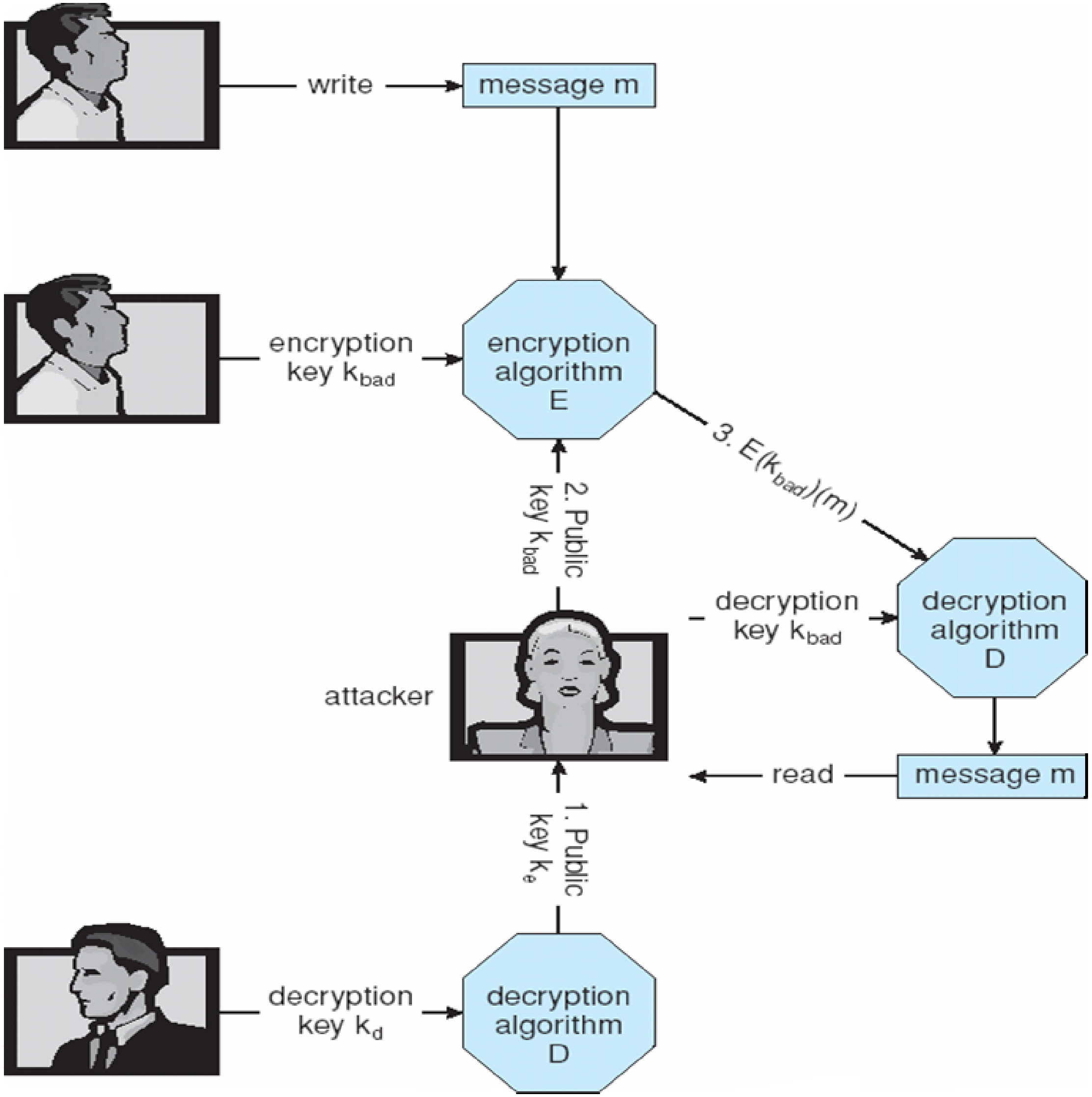
Authentication (Cont.)

- Why authentication if a subset of encryption?
  - Fewer computations (except for RSA digital signatures)
  - Authenticator usually shorter than message
  - Sometimes want authentication but not confidentiality
    - Signed patches et al
  - Can be basis for non-repudiation

Key Distribution

- Delivery of symmetric key is huge challenge
  - Sometimes done out-of-band
- Asymmetric keys can proliferate – stored on key ring
  - Even asymmetric key distribution needs care – man-in-the-middle attack

Man-in-the-middle Attack on Asymmetric Cryptography



Digital Certificates

- Proof of who or what owns a public key
- Public key digitally signed a trusted party
- Trusted party receives proof of identification from entity and certifies that public key belongs to entity
- Certificate authority are trusted party – their public keys included with web browser distributions
  - They vouch for other authorities via digitally signing their keys, and so on

End of Chapter 14