OPERATING SYSTEMS

SECURITY

SECURITY

In This Chapter:

- 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

SECURITY

SECURITY ISSUES:

External protection of a system. A classified site goes to extraordinary lengths to keep things physically tight. Among the issues to be considered:

Unauthorized access	Mechanism assuring only authorized individuals see classified materials.
Malicious	modification or destruction
Accidental	introduction of inconsistency.
Authentication	How do we know the user is who she says she is. Can have passwords on domains.

Protection of passwords is difficult. Issues include:

- It's very easy to guess passwords since people use simple and easily remembered words.
- Need exists to change passwords continually.
- Limiting number of tries before locking up.
- How to crack UNIX passwords.

SECURITY

Security Issues

Trojan Horse:	A piece of code that misuses its environment. The program seems innocent enough, however when executed, unexpected behavior occurs.
Trap Doors:	Inserting a method of breaching security in a system. For instance, some secret set of inputs to a program might provide special privileges.
Threat monitoring:	Look for unusual activity. Once access is gained, how do you identify someone acting in an unusual fashion?
Audit Log:	Record time, user, and type of access on all objects. Trace problems back to source.
Worms	Use spawning mechanism; standalone programs.
Internet Worm:	In the Internet worm, Robert Morse exploited UNIX networking features (remote access) as well as bugs in finger and sendmail programs. Grappling hook program uploaded main worm program.
Viruses	Fragment of code embedded in a legitimate program. Mainly effects personal PC systems. These are often downloaded via e-mail or as active components in web pages.
Firewall	A mechanism that allows only certain traffic between trusted and un-trusted systems. Often applied to a way to keep unwanted internet traffic away from a system.

SECURITY

Typical Security Attacks

ATTACK METHODS:

Attacks on a distributed system include:

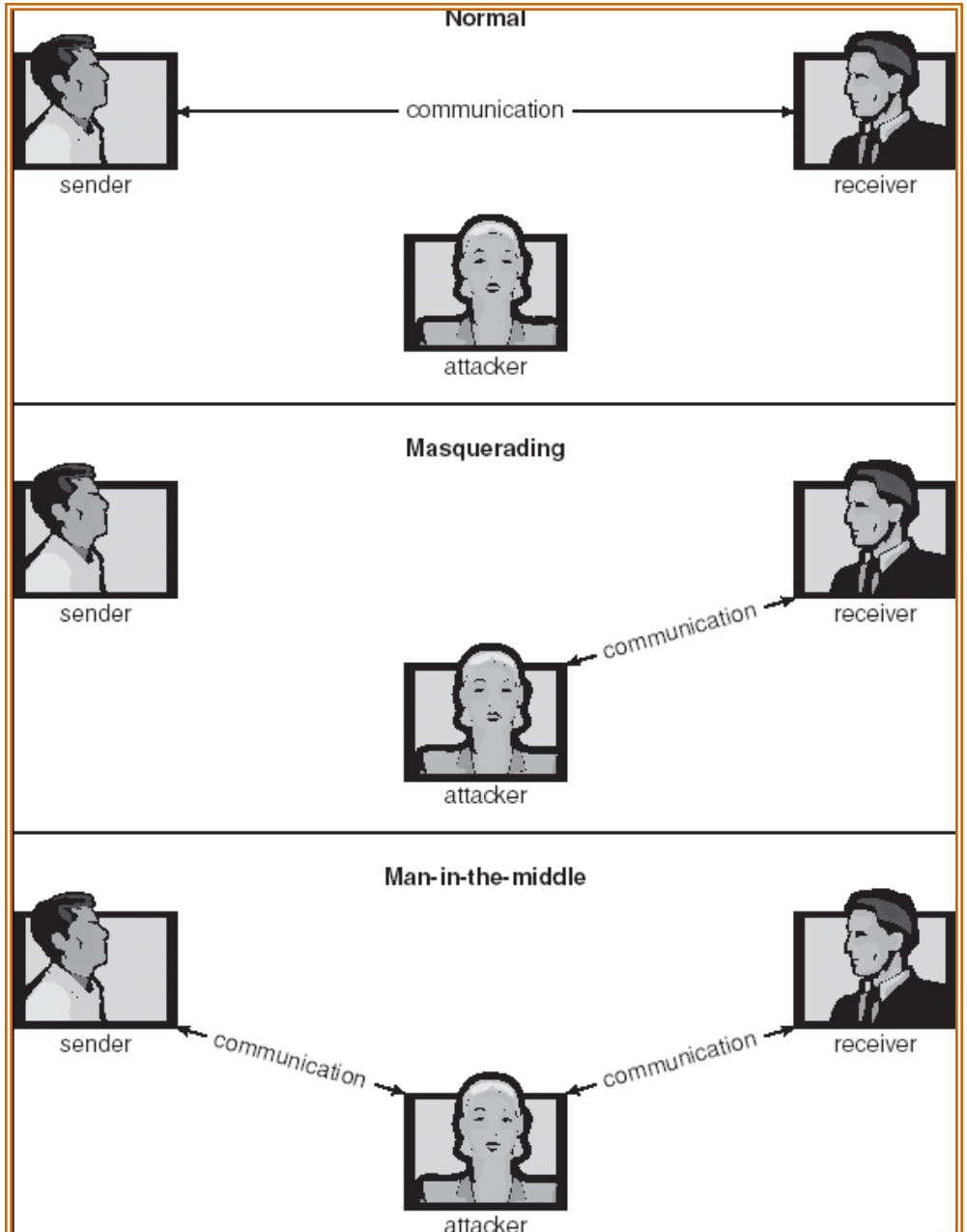
- Passive wiretapping. (unauthorized interception/reading of messages)
- Active wiretapping:

Modification	Changing a portion of the message.
Spurious messages	Introducing bogus messages with valid addresses and consistency criteria.
Site impersonation	Claiming to be some other logical node.
Replay	of previous transmission - repeating previous valid messages. (for example, authorization of cash withdrawal.)

SECURITY

ATTACK METHODS:

Typical Security Attacks



SECURITY

Typical Security Attacks

ATTACK METHODS:

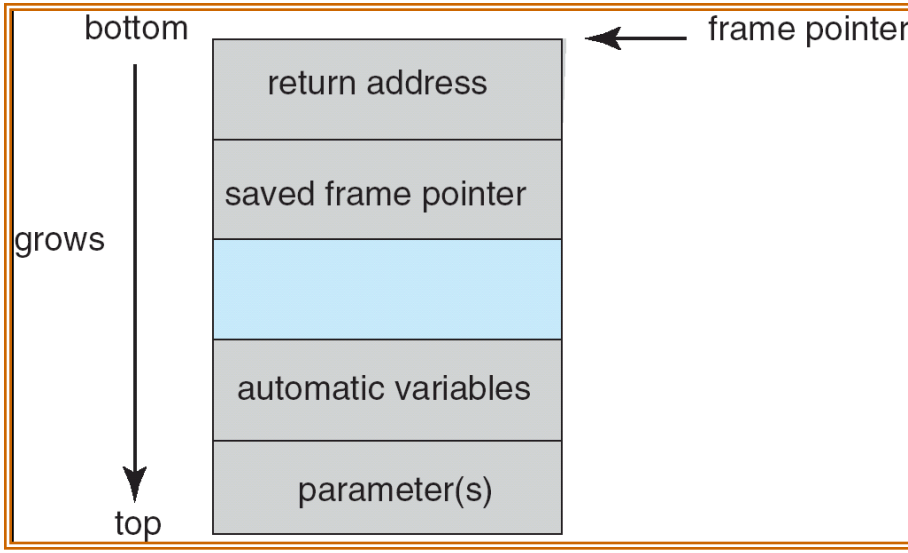
- Trojan Horse
  - Code segment that misuses its environment
  - Exploits mechanisms for allowing programs written by users to be executed by other users
  - **Spyware, pop-up browser windows, covert channels**
- Trap Door
  - Specific user identifier or password that circumvents normal security procedures
  - 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)

Typical Security Attacks

Example of Buffer Overflow Waiting To Happen:

```
#include <stdio.h>
#define BUFFER SIZE 256
int main(int argc, char *argv[])
{
   char buffer[BUFFER SIZE];
    int  other_data;

   if (argc < 2)
      return -1;
   else {
      strcpy(buffer,argv[1]);
      return 0;
   }
}
```



Typical Security Attacks

Viruses

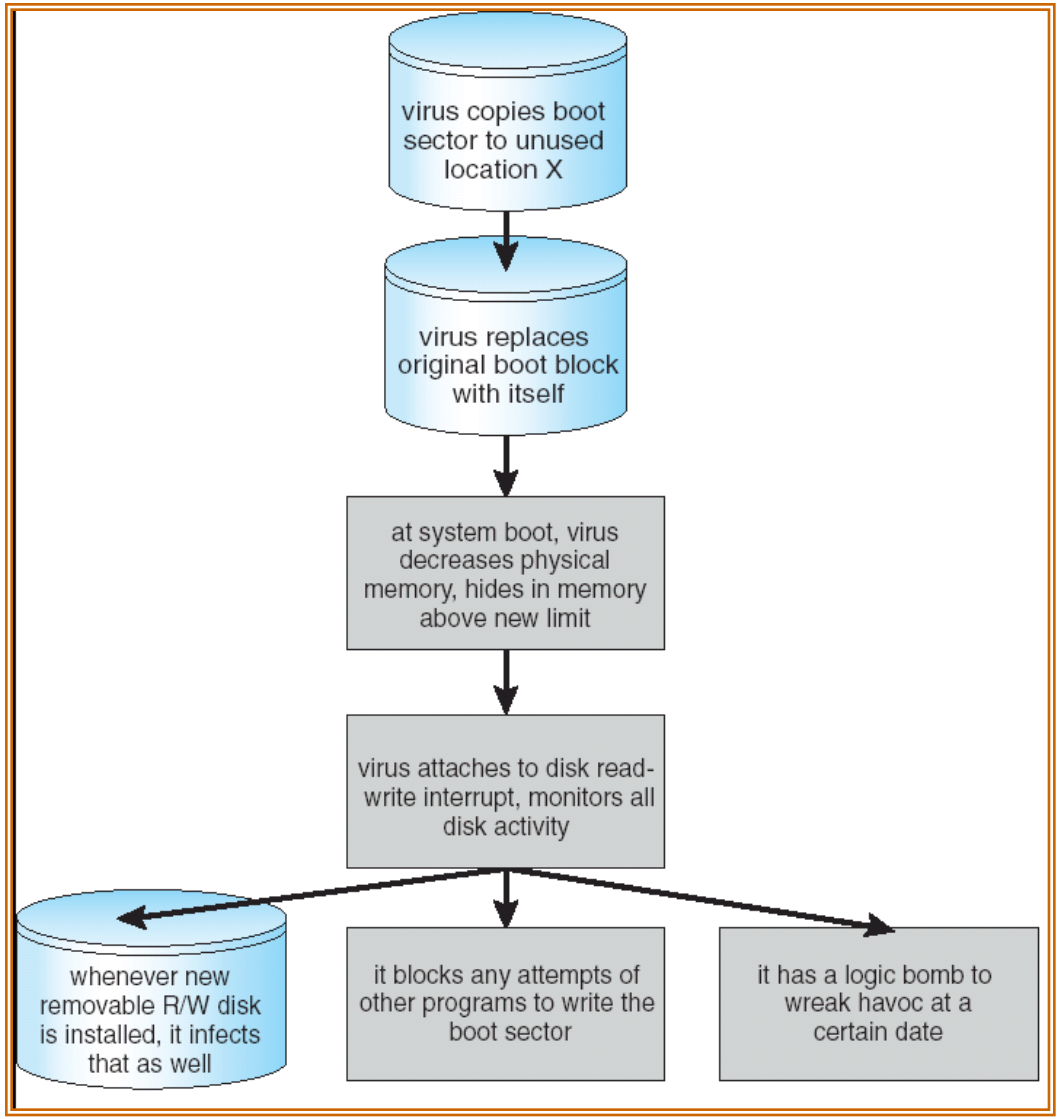
- Code fragment embedded in legitimate program
- Very specific to CPU architecture, operating system, applications
- Usually borne via email or as a macro
  - Visual Basic Macro to reformat hard drive

```
Sub AutoOpen()
Dim oFS
  Set oFS =
  CreateObject(''Scripting.FileSystemObject'')
  vs = Shell(''c:command.com /k format  c:'',vbHide)
End Sub
```

SECURITY

Typical Security Attacks

A Boot Sector Virus

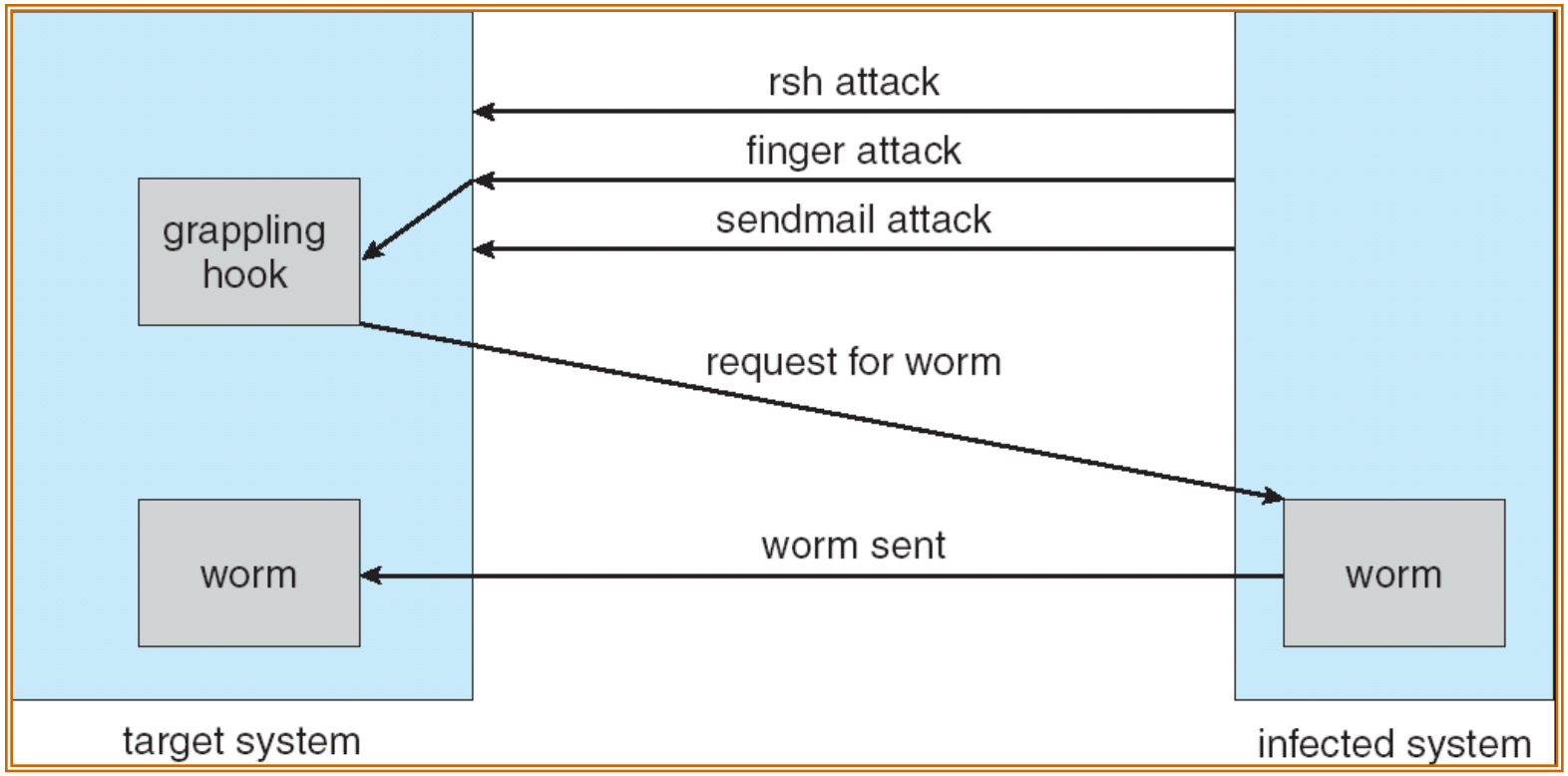


Typical Security Attacks

System And Network Threats

- Worms – use **spawn** mechanism; standalone program
- Internet worm
  - Exploited UNIX networking features (remote access) and bugs in *finger* and *sendmail* programs. (See next slide)
  - **Grappling hook** program uploaded main worm program
- Port scanning
  - Automated attempt to connect to a range of ports on one or a range of IP addresses
- Denial of Service
  - Overload the targeted computer preventing it from doing any useful work
  - Distributed denial-of-service (**DDOS**) come from multiple sites at once

Morris Internet Worm



SECURITY

Cryptography

DEFINITIONS:

Encryption:

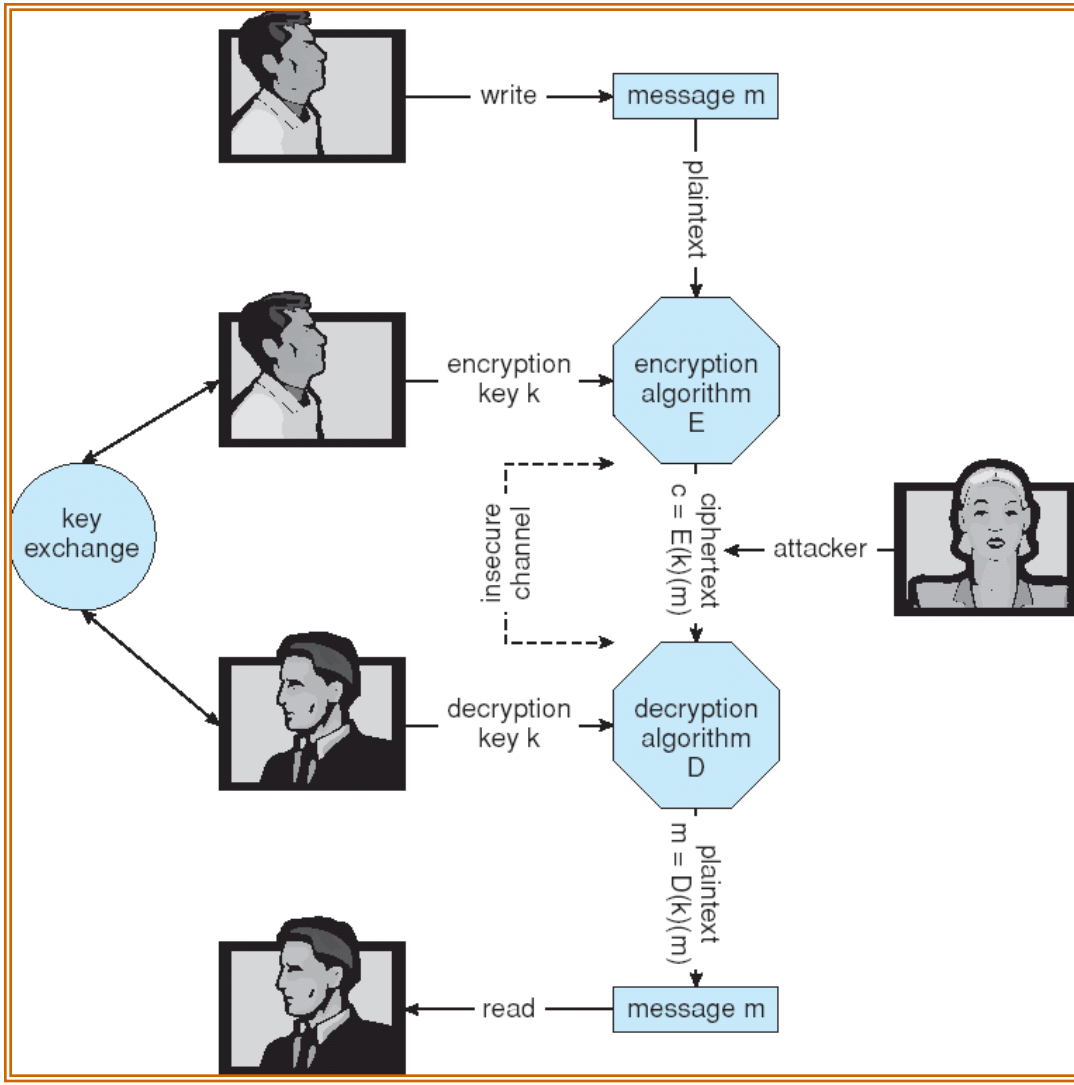
$$C = E(M, Ke)$$

E = Encyphering Algorithm
M = Message - plain text
Ke = Encryption key
C = Cyphered text

Decryption:

$$M = D(C, Kd)$$

D = Decyphering Algorithm
Kd = Decryption key



SECURITY

Cryptography

DEFINITIONS:

Cryptosystems are either Conventional or Public Key

- Conventional is symmetric; Ke = Kd , so the key must be kept secret. Algorithms are simple to describe, but complex in the number of operations.
- Public key is asymmetric; Ke != Kd , so Ke can be made public. Kd is secret and can't easily be derived from Ke .

Security against attack is either:

- **Unconditionally secure** - Ke can't be determined regardless of available computational power.
- **Computationally secure:** - calculation of Kd is economically unfeasible (it would overwhelm all available computing facilities.)

The only known unconditionally secure system in common use!

- Involves a random key that has the same length as the plain text to be encrypted.
- The key is used once and then discarded. The key is exclusively OR'd with the message to produce the cypher.
- Given the key and the cypher, the receiver uses the same method to reproduce the message.

SECURITY

Data Encryption Standard

DATA ENCRYPTION STANDARD (DES):

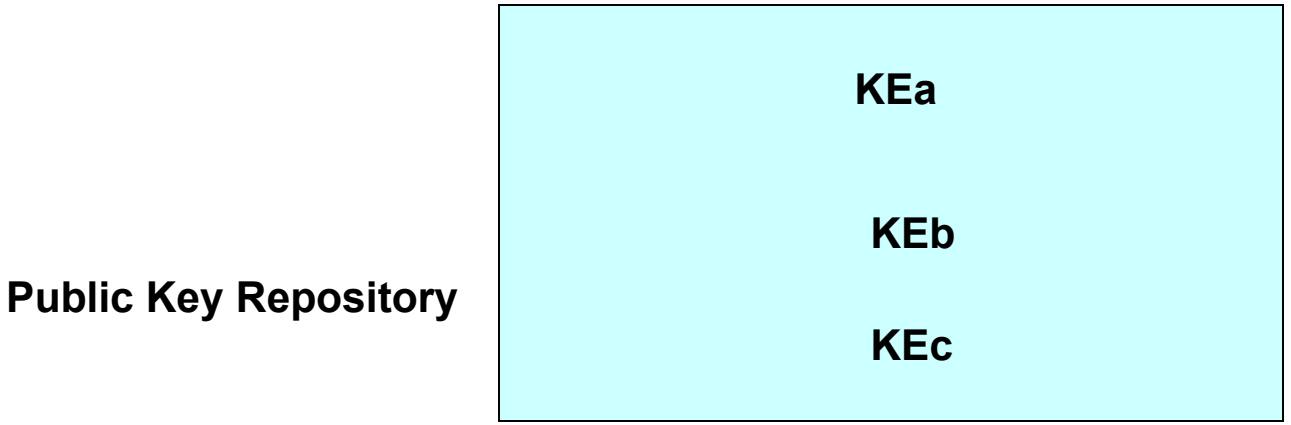
- The official National Institute of Standards and Technology (NIST), (formerly the National Bureau of Standards) encryption for use by Federal agencies.
- The source of security is the non-linear many-to-one function applied to a block of data. This function uses transposition and substitution. The algorithm is public, but the key (56 bits) is secret.
- Computational power today can crack a 56 bit code.
- In common use today is Triple DES in which 3 different keys are used, making the effective key length 168 bits.

SECURITY

Public Key Cryptosystems

The general principle is this:

1. Any **RECEIVER A** uses an algorithm to calculate an encryption key **KEa** and a decryption key **KDa**.
2. Then the receiver PUBLICIZES **KEa** to anyone who cares to hear. But the receiver keeps secret the decryption key **KDa**.
3. **User B** sends a message to **A** by first encrypting that message using the publicized key for that receiver **A**, **KEa**.
4. Since only **A** knows how to decrypt the message, it's secure.



SECURITY

Public Key Cryptosystems

To be effective, a system must satisfy the following rules:

a) Given plaintext and ciphertext, the problem of determining the keys is computationally complex.

b) It is easy to generate matched pairs of keys Ke, Kd that satisfy the property

D(E(M, Ke), Kd) = M.

This implies some sort of trapdoor, such that Ke and Kd can be calculated from first principles, but one can't be derived from the other.

c) The encryption and decryption functions E and D are efficient and easy to use.

d) Given Ke , the problem of determining Kd is computationally complex.

What is computationally difficult? Problems that can't easily be calculated in a finite time.

Examples include: factoring the product of two very large prime numbers; the knapsack problem.

These problems are NP complete - solution times are exponential in the size of the sample.

SECURITY

Public Key Cryptosystems

To be effective, a system must satisfy the following rules:

e) For almost all messages it must be computationally unfeasible to find ciphertext key pairs that will produce the message.

(In other words, an attacker is forced to discover the true (M,Ke) pair that was used to create the ciphertext C.)

f) Decryption is the inverse of encryption.

E(D(M, Kd), Ke) = D(E(M, Ke), Kd)

Public Key Cryptosystems

AN EXAMPLE:

1. Two large prime numbers p and q are selected using some efficient test for primality. These numbers are secret:

   Let p = 3, q = 11

2. The product n = p * q is computed.

   n = 3 * 11 = 33.

3. The number Kd > max(p, q) is picked at random from the set of integers that are relatively prime to and less than L(n) = (p - 1) (q - 1).

   L(n) = (p - 1) (q - 1) = 20.
   Choose Kd > 11 and prime to 20.
   Choose Kd = 13.

4. The integer Ke , 0 < Ke < L(n) is computed from L(n) and Kd such that Ke * Kd = 1 (mod L(n)).

   0 < Ke < 20
   Ke = 17. (since 17 * 13 = 221 = 1 (mod 20))

SECURITY

Public Key Cryptosystems

AN EXAMPLE:

Separate the text to be encoded into chunks with values 0 - (n - 1).

In our example, we'll use < space = 0, A = 1, B = 2, C = 3, D = 4, E = 5 >.

Then " B A D <sp> B E E " --> "21 04 00 25 05"

21 ^ 17	(mod 33)	=	21.	21 ^ 13	(mod 33)	=	21.
04 ^ 17	(mod 33)	=	16.	16 ^ 13	(mod 33)	=	04.
00 ^ 17	(mod 33)	=	00.	00 ^ 13	(mod 33)	=	00.
25 ^ 17	(mod 33)	=	31.	31 ^ 13	(mod 33)	=	25.
05 ^ 17	(mod 33)	=	14.	14 ^ 13	(mod 33)	=	05.

This whole operation works because, though n and Ke are known, p and q are not public. Thus Kd is hard to guess.

[Note: recently a 100 digit number was successfully factored into two prime numbers.]

SECURITY

AUTHENTICATION AND DIGITAL SIGNATURES:

Sender Authentication:

In a public key system, how does the receiver know who sent a message (since the receiver's encryption key is public)?

Suppose **A** sends message **M** to **B**:

a) **A** DECRYPTS **M** using **A's** **Kd(A)** .
b) **A** attaches its identification to the message.
c) **A** ENCRYPTS the entire message using **B's** encryption, **Ke(B)**

$$C = E((A, D(M, Kd(A))), Ke(B))$$

d) **B** decrypts using its private key **Kd(A)** to produce the pair **A,** **D(M, Kd(A)).**
e) Since the proclaimed sender is **A**, **B** knows to use the public encryption key **Ke(A)**.

Capture/Replay

In this case, a third party could capture / replay a message.

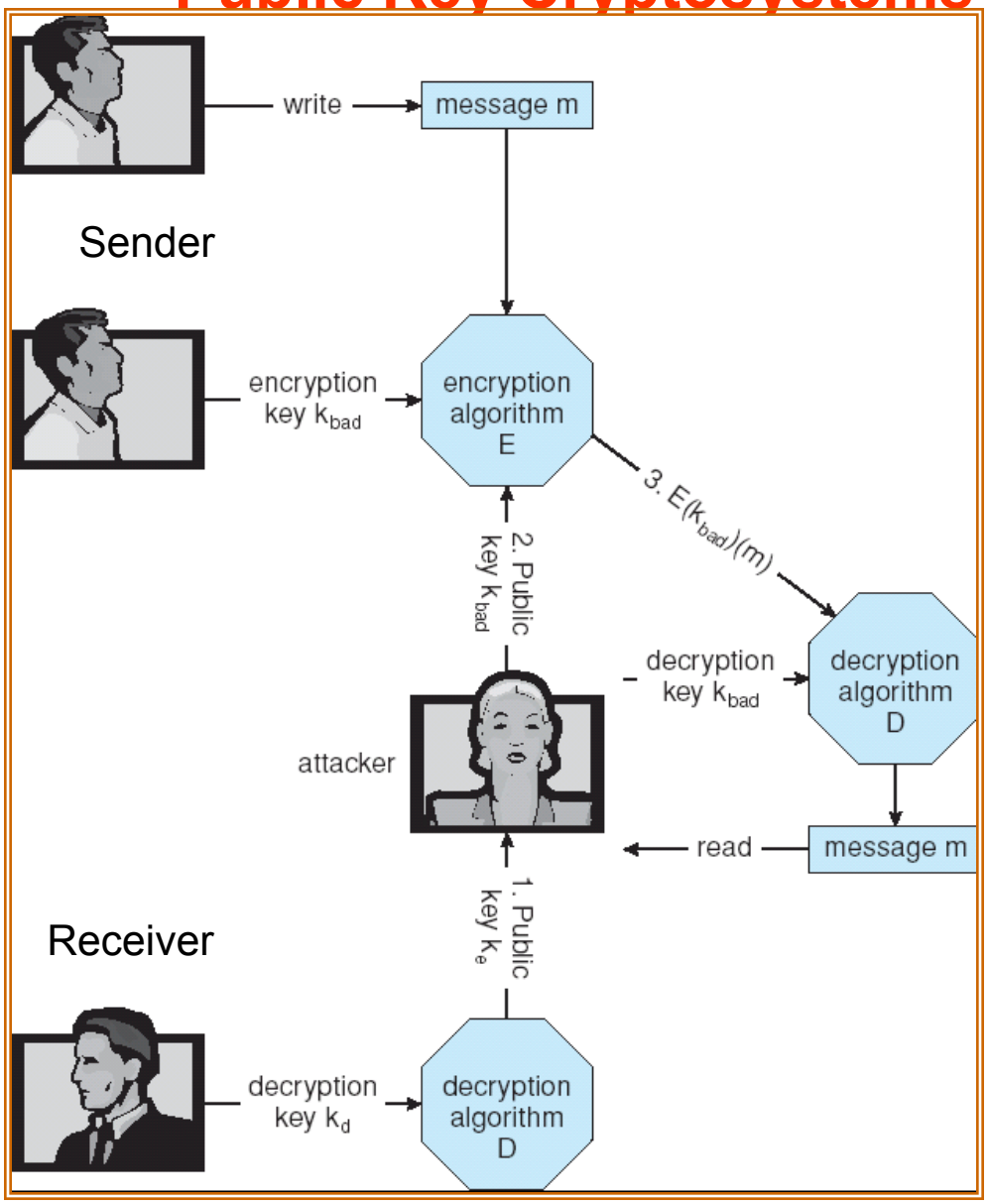
The solution is to use a rapidly changing value such as time or a sequence number as part of the message.

SECURITY

Man-in-the-middle Attack on Asymmetric Cryptography

Here are the attack steps for this scenario:

1. Sender wishes to send a message to Receiver.
2. S asks R for its encryption key.
3. When R returns key, that key is intercepted by the attacker who substitutes her key.
4. Sender encrypts message using this bogus key and returns it.
5. Since the attacker is the owner of this bogus key, the attacker can read the message.



SECURITY

Example - SSL

- Insertion of cryptography at one layer of the ISO network model (the transport layer)
- SSL – Secure Socket Layer (also called TLS)
- Cryptographic protocol that limits two computers to only exchange messages with each other
  - Very complicated, with many variations
- Used between web servers and browsers for secure communication (credit card numbers)
- The server is verified with a **certificate** assuring client is talking to correct server
- Asymmetric cryptography used to establish a secure **session key** (symmetric encryption) for bulk of communication during session
- Communication between each computer uses symmetric key cryptography

SECURITY

Example – Windows XP

- Security is based on user accounts
  - Each user has unique security ID
  - Login to ID creates **security access token**
    - Includes security ID for user, for user's groups, and special privileges
    - Every process gets copy of token
    - System checks token to determine if access allowed or denied
- Uses a subject model to ensure access security. A subject tracks and manages permissions for each program that a user runs
- Each object in Windows XP has a security attribute defined by a security descriptor
  - For example, a file has a security descriptor that indicates the access permissions for all users

SECURITY

Security Classifications

U.S. Department of Defense outlines four divisions of computer security: **A**, **B**, **C**, and **D**.

- **D** – Minimal security.
- **C** – Provides discretionary protection through auditing. Divided into **C1** and **C2**. **C1** identifies cooperating users with the same level of protection. **C2** allows user-level access control.
- **B** – All the properties of **C**, however each object may have unique sensitivity labels. Divided into **B1**, **B2**, and **B3**.
- **A** – Uses formal design and verification techniques to ensure security.

SECURITY

Wrap Up

In this chapter we've looked at how to secure information that may be placed in hazardous public forums.

Data on the net is an excellent example here.