Operating Systems ICS 431

Week 15

Ch.: Security

Dr. Tarek Helmy

Ch 15: Security

- Objectives of this chapter:
  - To discuss security threats and attacks.
  - To explain the fundamentals of authentication and authorization.
  - To discuss various security attacks.
  - To examine the uses of cryptography in computing.

The Security Problem

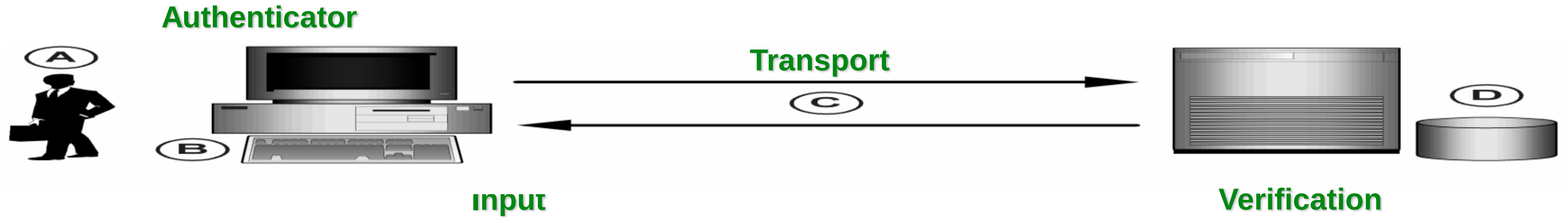
- The system is secure if its resources are utilized and accessed as intended under all conditions.
- This is may not be possible due to different ways of security violations.
- **Why security**:
  - To protect the system from malicious/intended misuse, i.e.:
    - Unauthorized access.
    - Unauthorized modification or destruction of data.
  - Many computer systems contain very important information:
    - Financial information, Software in development, Sensitive data
- Security considers external environment of the system, and protects the system from intended attacks attempt to use its resources.
- **Threat** is a potential security violation.
- **Attack** is an attempt to violate security.
  - Attacks can be accidental or malicious/Intentional/planned.

Security Levels

- To secure the system, we must take security measures at four levels:
  - Physical: The site or sites containing the computer system must be secured against the entry of intruders.
  - Human: Users must be screened carefully to reduce the chance of authorizing a user who gives access to an intruder.
  - Network: As data travels over network lines, so data should be protected from the attack.
  - OS: The operating system must protect the machine from intentional or focused security gaps.
  - **We will discuss the security at the OS level**.

Authentication and Authorization

- Authentication is the process of **validating/identifying the users or processes.**
- Authorization is the process that decides whether users/processes are permitted to perform the functions they request or not.
- **Authorization** is not performed until the user/process is **authenticated.**
- **Authentication and Authorization used to** verify that?
    - Something I know (that nobody else should know)
    - Something I have (that nobody else should have)
    - Something I am (that nobody else should be…)
- How do we ensure that only authorized users/processes access their resources?
    - OS uses **Access Control Matrix** to set read/write/execute privileges on objects.
- Methods of authenticating users are:
    - ID and Passwords (knowledge)
    - Keys or digital/smart cards (physical means)
    - Biometrics, like eye tracking or finger print (physical attribute)
- User identity most often established through **ID** and **passwords**, which is considered a special case of either keys or capabilities.



Threats Against Authentication

I want to pretend to be you:

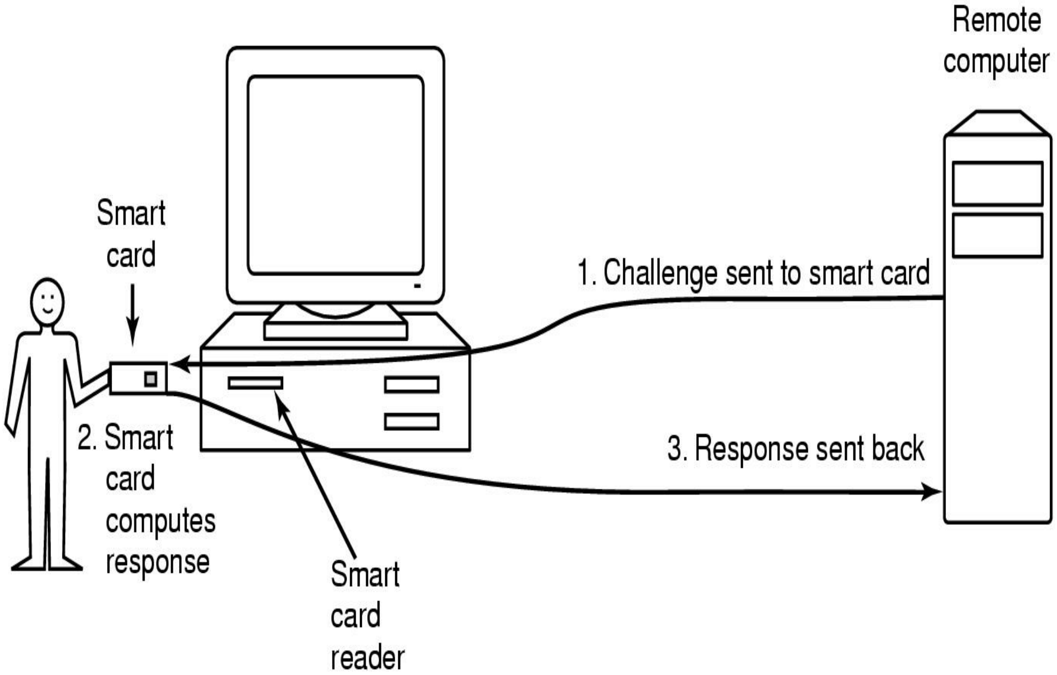
- Password can be guessed
  - Particularly some one or some programs can guess your username and password.
- Password can be captured.
  - From the sticky note on your monitor or on a list in your desk.
  - By monitoring your communications or looking over your shoulder.
  - By getting between you and the system you are talking to.
  - By using some SW that records every keystroke on your PC and then view them.
  - By using the key manager utility program “rundll32.exe/keymgr.dll”." that will launch and display a list of passwords used on the computer.

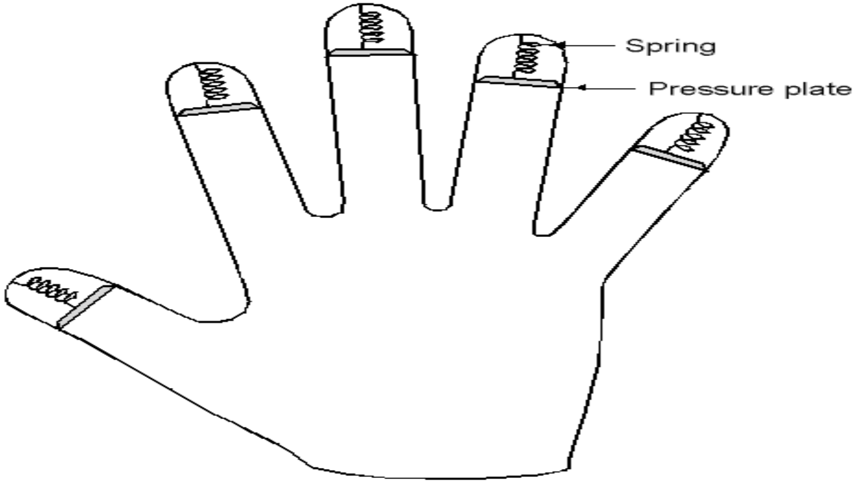
How to Avoid Passwords Capturing?

- **Passwords must be kept secret.**
  - Frequent change of passwords. Some system force the user to change the PW at regular intervals.
  - Use of "non-guessable" passwords. Long PW with a combination of letters and numbers is recommended.
    - Methods may be used to make this harder but that makes it harder on users to remember passwords
      - If it is too difficult to remember a password, then writing it down on paper is a major security problem.
  - Do not choose to save a password in an application or on a Web site.
  - **FYI, programs could be used to discover a password by trying all words in a dictionary or all combinations of letters up to 10 in length, etc**.
- Passwords may be encrypted. Only encoded PW are stored and the given PW will be encoded first and then compared with the stored one.
- To avoid the problem of PW sniffing, a system may allow the PW to be associated with the PIN (usually used only once). The system asks the user to enter the PIN and then get the current PW that may be generated based on the current time for instance.

Authentication using a Physical Object or Biometrics

- Physical characteristics
  - Fingerprint
  - Hand Geometry
  - Face Geometry
  - Voice Print
  - Palm print
  - DNA
- Magnetic cards
  - Magnetic stripe cards
  - Chip cards:
  - Smart cards



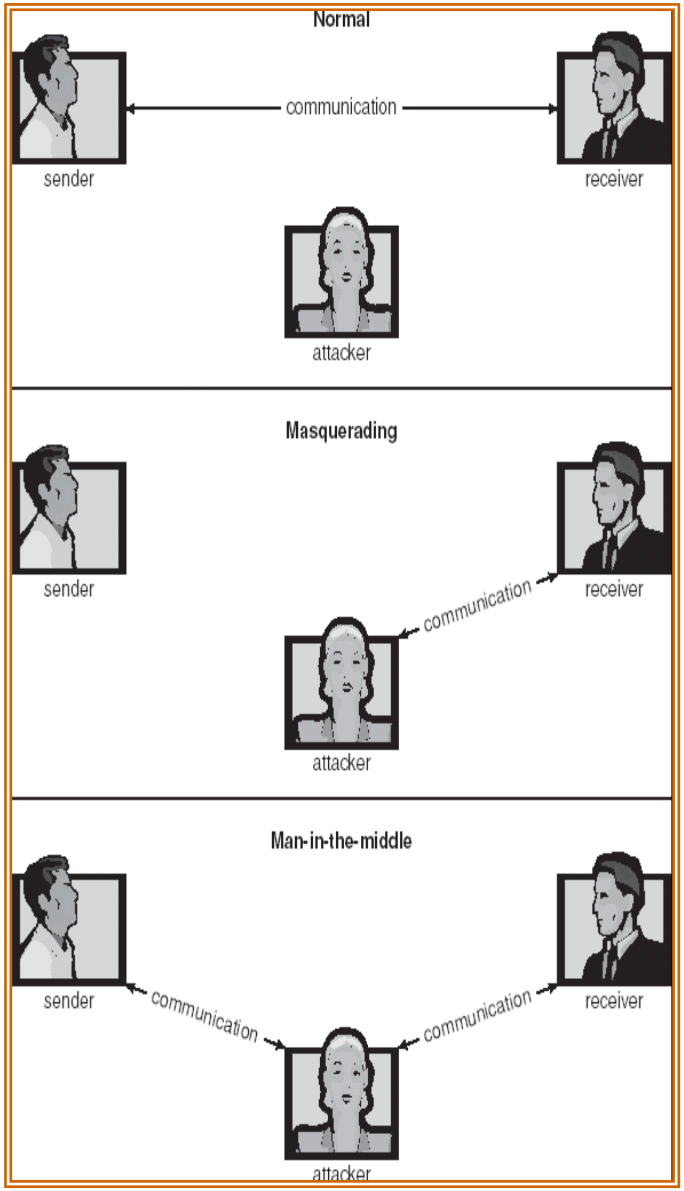


Security Violation Categories

- **Security Violation Categories are:**
  - Violation of confidentiality: Involves unauthorized reading of data, capturing secret data from the system.
  - Violation of integrity: Involves unauthorized modification of data.
  - Violation of availability: Involves unauthorized destruction of data.
  - Stealing of service: Involves unauthorized use of resources.
  - Denial of service: Involves preventing useful use of the system.

Methods of Attacking

- Masquerading/hidden (violate the authentication): The attacker pretends to be an authorized user of a system in order to gain more privileges than he is authorized.
  - i.e. If a legal user leaves the session open and a co-worker may act as a masquerade attacker.
- Man-in-the-middle attack: An attacker sits in the data flow of communication.
  - i.e. he/she can inject false information and intercept the data transferred between them.
- Replay attack: Repeat of a valid data with message modification.
- Login Spoof
  - Creating a login screen as a process on a public machine looks exactly like a real one.
  - You log into that process which records your user ID and password.
  - Result: Getting between you and the system without your knowledge and stole your user ID and password.



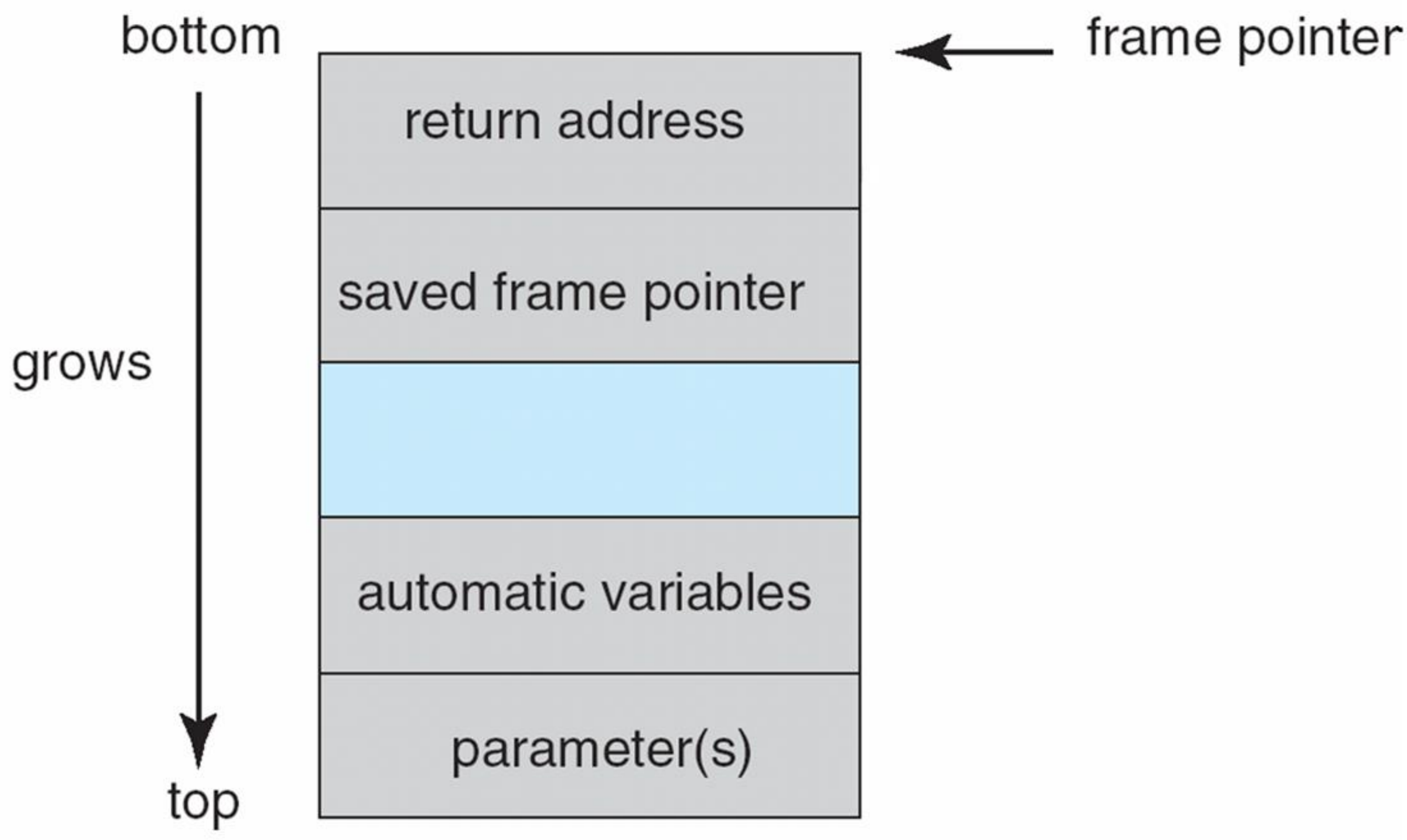
Program Threats in OS

- A program written by one user and used by another may cause unexpected behavior.
- Here are the common methods by which programs cause security violation.
- **Trojan Horse**
  - It is a code segment that misuses its environment.
  - Appear to have one function, but do something different.
  - Once running, may do something unexpected such as delete, copy or move files.
  - Because the program is being run by the user, the program has the rights of the user and can perform OS activities without additional authorization
  - A Trojan horse program could be used to gain access to files of another user on a shared system or could change the user's file permissions to make a file readable by all users.
  - **Examples of Trojan horse:**
  - **Trojan-Downloader:** can download and install new versions of malicious programs onto your computer.
  - **Trojan-Banker**: Its purpose is to steal your account data for online banking systems.
  - **Trojan-Mail-finder**: This robs email addresses from your endpoint.
  - **Trojan-DoS:** This Trojan can start up the Denial of Service (DoS) attacks.

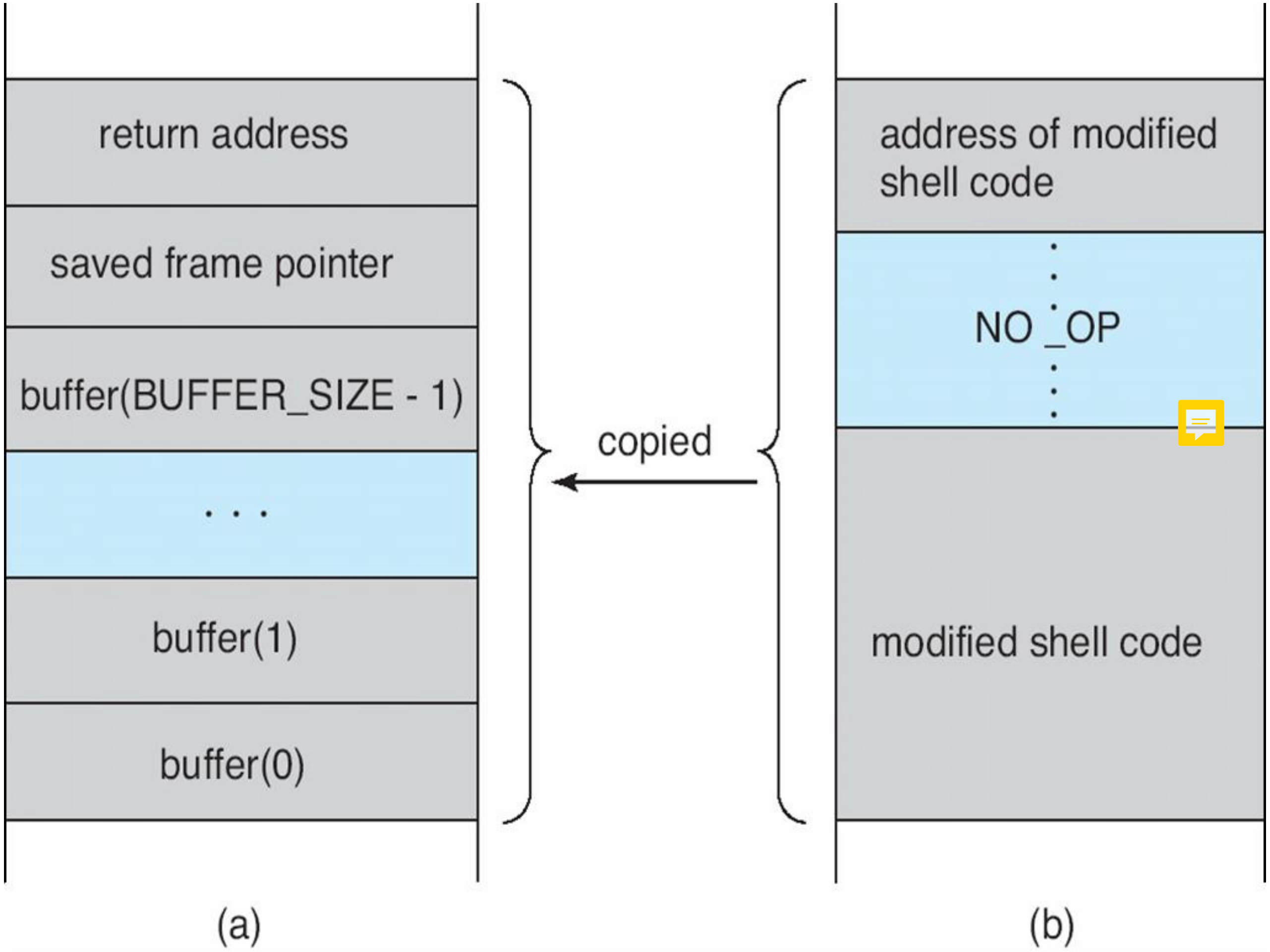
Program Threats in OS

- **Trap Door**: is a secret entry point within a program, specially designed by the program's creator for his use only.
  - Trap door can be easily exploited if an unauthorized user becomes aware of it.
- **Logic Bomb**: initiates a security hole under certain conditions when a predefined set of parameters met.
  - It would be hard to detect because under normal operation, there would be no holes.
- **Stack and Buffer Overflow**:
  - The stack or buffer-over-flow attack is the most common way for an attacker outside of the system to gain unauthorized access to the system.
  - By trial and error or by examination of the source code, **the attacker determines the weakness of the program and writes a program to do the following:**
    - Overflow an input field, until it writes into a stack.
    - If input variable's length is not checked, inserted code 'leaks' from buffer to execution stack.
    - Overwrite the current return address on the stack with the address of the code the attacker wishes to execute.
    - Write a simple code for the next space in the stack.

Layout of Typical Stack Frame



Hypothetical Stack Frame



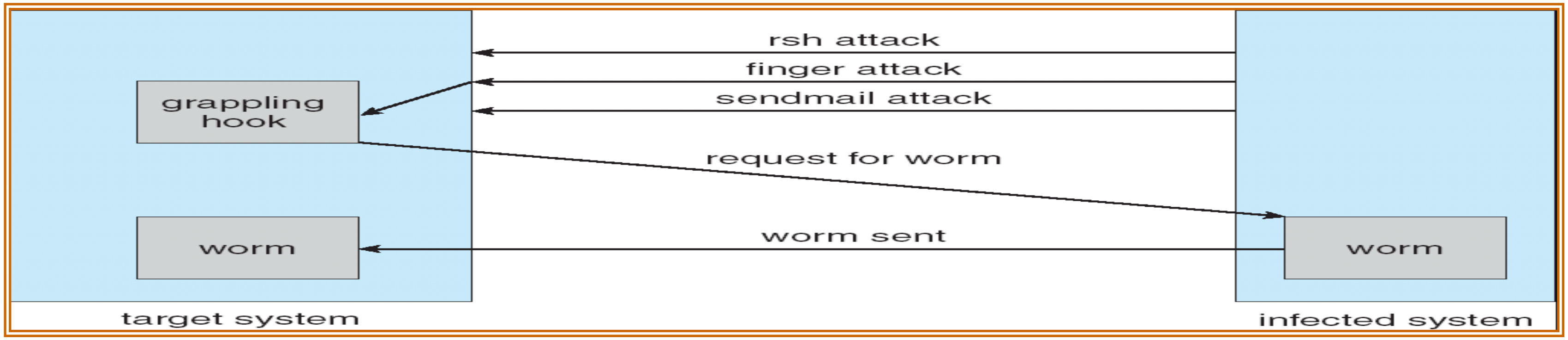
Before attack

After attack

System Threats

- Due to the fact that most OSs allow processes to generate other processes.
- This creates situations in which OS resources and files are misused.
- Two common methods for **achieving this misuse are:** **worms and viruses**.

- **Internet worm** (Robert Morris, 1988, first year graduate student, Cornell Uni.)
  - A worm is self-propagating pieces of code that reduce system performance by spawning/generating copies of themselves repeatedly.
  - **Worm is a complete stand alone program**.
  - Spreads through e-mail
  - **May not do harm, but dragged down many machines**.
  - The worm was made up of two programs, grappling hook (bootstrap) and the main program.
    - **The grappling program** run on the machine it attacks and connected to the original machine to upload a copy of the main worm onto the hooked system.

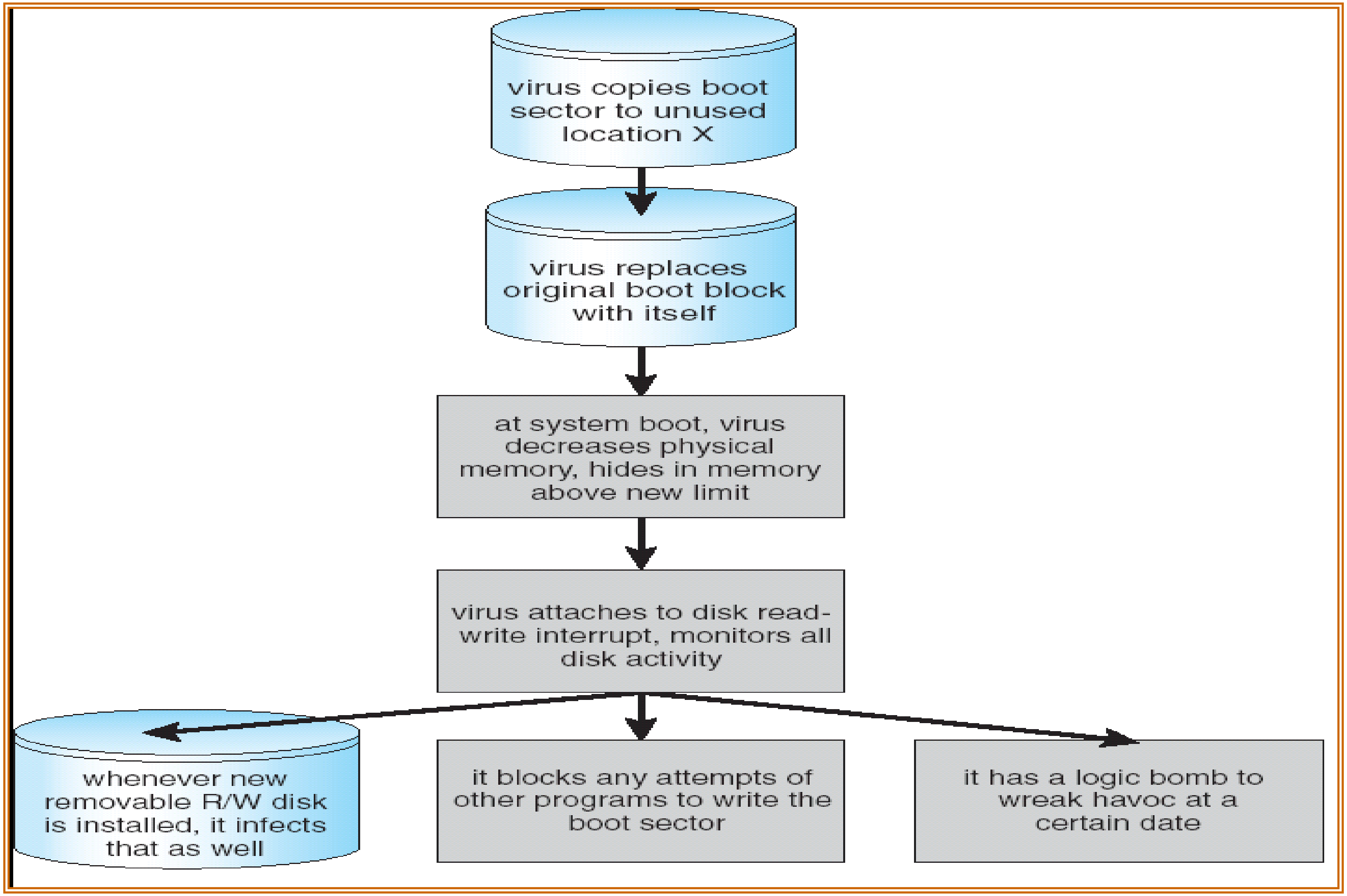


System Threats

- **Viruses**
  - Virus is a fragment of code embedded inside a program
  - Usually spread by users downloading viral programs from public sources or exchange disks.
- Virus types according to their “homes”:
  - Boot sector viruses
  - Memory-resident viruses
  - Macro viruses
    - A virus that is written in a macro language and placed within a document.
    - When the document is opened and the macro is executed, commands in the macro language do the destruction.
    - Usually causes a sequence of actions to be performed automatically when the application is started.
  - Example of modifying or destroying of data files:
    - ✓ Add code to end of program
    - ✓ Replace first line (entry point) with jump to this code, end of virus jumps to real entry point
    - ✓ Virus has control, tries to replicate, also does malicious things
- **Visual Basic Macro that reformats the hard drive**

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

A Boot-sector Computer Virus


virus copies boot sector to unused location X
virus replaces original boot block with itself
at system boot, virus decreases physical memory, hides in memory above new limit
virus attaches to disk read-write interrupt, monitors all disk activity
whenever new removable R/W disk is installed, it infects that as well
it blocks any attempts of other programs to write the boot sector
it has a logic bomb to wreak havoc at a certain date

System Threats

- **Denial of Service**
  - The focus of this attack is to **disable the legal use of a system rather than gaining information or stealing resources**.
  - Overload the targeted computer preventing it from doing any useful work.

- Denial of service attacks on a local machine
  - Too many processes
  - Too many files or files too big
  - Too much of CPU time in looping.

- Denial of service attacks on a network
  - Service flooding - **too many requests**
  - Downloading Java scripts that use all available CPU times.
  - Message flooding
  - Starting several TCP connections that eat up all the network resources.
  - Email attacks

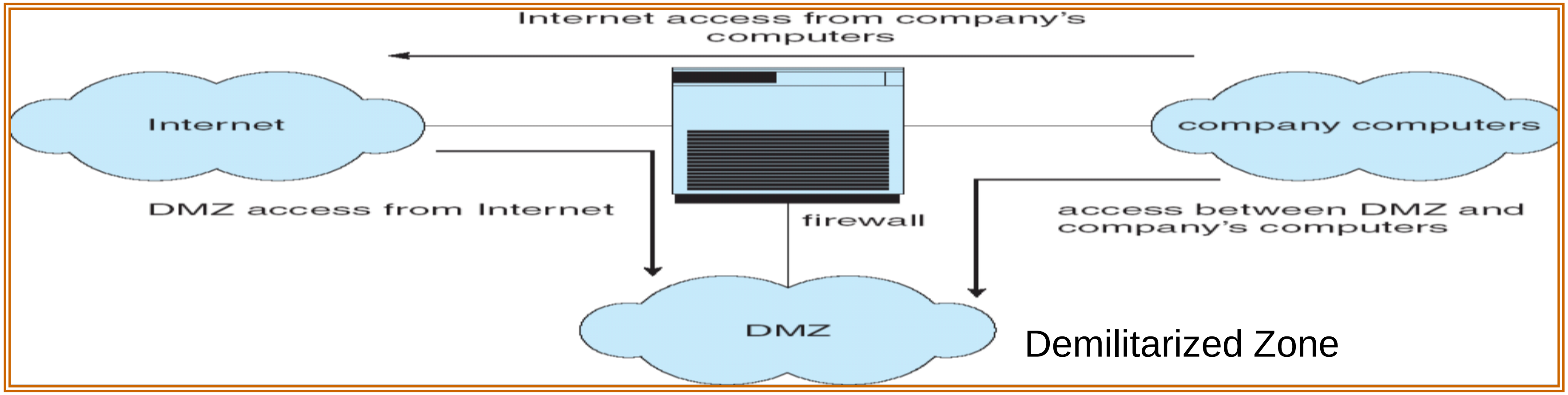
Protection against System Threats

- Protection against system threats:
  - Don’t download software from unknown sources
  - Don’t share removable memory
  - Check hard disk at boot up every time
  - Check every flash memory as its inserted
  - Use antiviral programs to find and remove viruses
  - Keep Your Anti-Virus Software Up to Date.
  - Run Regularly Scheduled Scans with Your Anti-Virus Software.
  - Keep Your Operating System Current.
  - Secure Your Network.
  - Think Before You Click.
- Check-Sum feature
  - The OS may add up all bits in a program and save it.
  - If the program ever changes in size, it is probably because of a virus installed in it.

Implementing Security Defenses

- Check for suspicious patterns of activity: Several incorrect password attempts may signal password guessing.
- Audit log: Records the time, user, and type of all accesses to an object; this is useful for recovery from a violation and developing better security measures.
- Intrusion detection endeavors to detect attempted or successful intrusions
  - **Signature-based** detection spots known bad patterns
  - **Anomaly detection** spots differences from normal behavior
    - Can detect **zero-day** attacks
  - **False-positives** and **false-negatives** problems
- Virus protection
- Scan the system periodically for security holes: Done when the computer is relatively unused.
  - Unauthorized programs in system directories
  - Unexpected long-running processes
  - Improper directory protections
  - Improper protections on system data files
  - Dangerous entries in the program search path

Network Security Through Domain Separation Via Firewall


Internet access from company's computers
Internet
company computers
firewall
DMZ access from Internet
access between DMZ and company's computers
DMZ

Demilitarized Zone

- **DMZ (Demilitarized Zone)** is to ensure that publicly accessible servers cannot contact other internal network segments, in the event that a server is compromised.
- A firewall is 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.**
- Can be implemented in software or hardware.
- **Personal firewall** is software layer on a given host
  - Can monitor/limit traffic to and from the host
- **Application proxy firewall** understands application protocol and can control them (i.e. SMTP)
- **System-call firewall** monitors all important system calls and apply rules to them (i.e. this program can execute that system call)

Encryption Scheme

- **Encryption** is the process of converting the original data into another form which cannot be easily understood by anyone except authorized parties.
- Encryption systems include 4 main components:
  - **Plaintext (Message):** the unencrypted message
  - Encryption algorithm: the function used to encrypt the message.
  - **Key:** is a string of bits used by a **cryptographic** algorithm to transform plain text into cipher text or vice versa.
    - For example, let say that you wanted to encrypt the message, 'I like KFUPM', using the key, 'hello'. Then the resulting encrypted message might look like, 'KJSUTSQKJIN'. Now, if we introduce another key, say 'sunny', and encrypt the same message, we might get something different.
  - **Ciphertext:** is the encrypted message produced by the encryption function.
  - **Decryption:** is the reverse of encryption. It doesn't always use the same key or algorithm. Plaintext results from decryption.

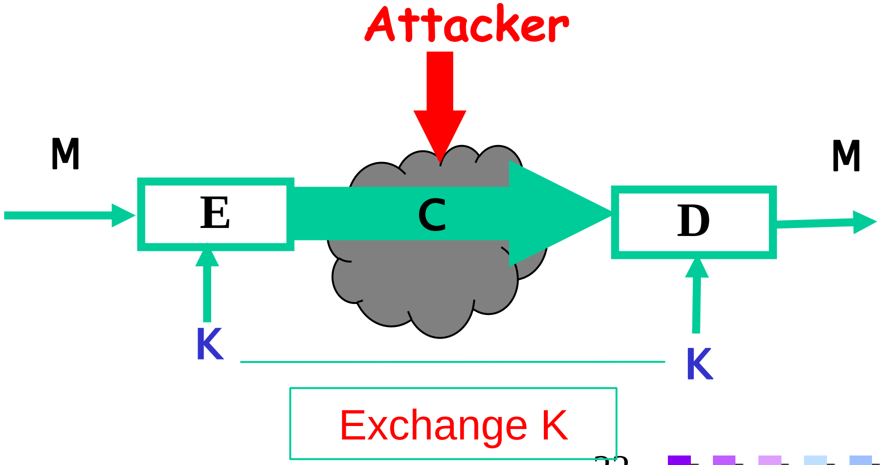
Key …………… K

Message ………. M

Cipher-text ………**C**

Encryption Algorithm .. **E**

Decryption Algorithm ….. **D**



Two kinds of Cryptography

Symmetric

1. Symmetric-key cryptography is based on sharing the secret key.
2. A and B **agree on the encryption algorithm & the secret-key.**
3. The secret key can either be a number, a word or a string of random letters.
4. A takes his plaintext message and encrypts it using the encryption algorithm and the key.
5. This creates a ciphertext message.
6. A sends the ciphertext message to B.
7. B decrypts the ciphertext message with the same algorithm and key and reads it.

Asymmetric

1. Asymmetric-key cryptography is based on **public-key** and **private-keys**.
2. It uses public and private keys to encrypt and decrypt data.
3. A public-key is made freely available to anyone who might want to send you a message. The second private-key is kept a secret so that you can only know.
4. When someone wants to send an encrypted message, they can pull the intended recipient's public key from a public directory and use it to encrypt the message before sending it.
5. The recipient of the message can then decrypt the message using his/her related private key.
6. On the other hand, if the sender encrypts the message using his/her private key, then the message can be decrypted only using that sender's public key, thus authenticating the sender.

Symmetric: Secure Communication over Insecure Medium

Encryption:

C = E(M, Ke)

E = Encryption Algorithm

M = Message - plain text

Ke = Encryption key

C = Cipher/secured Text

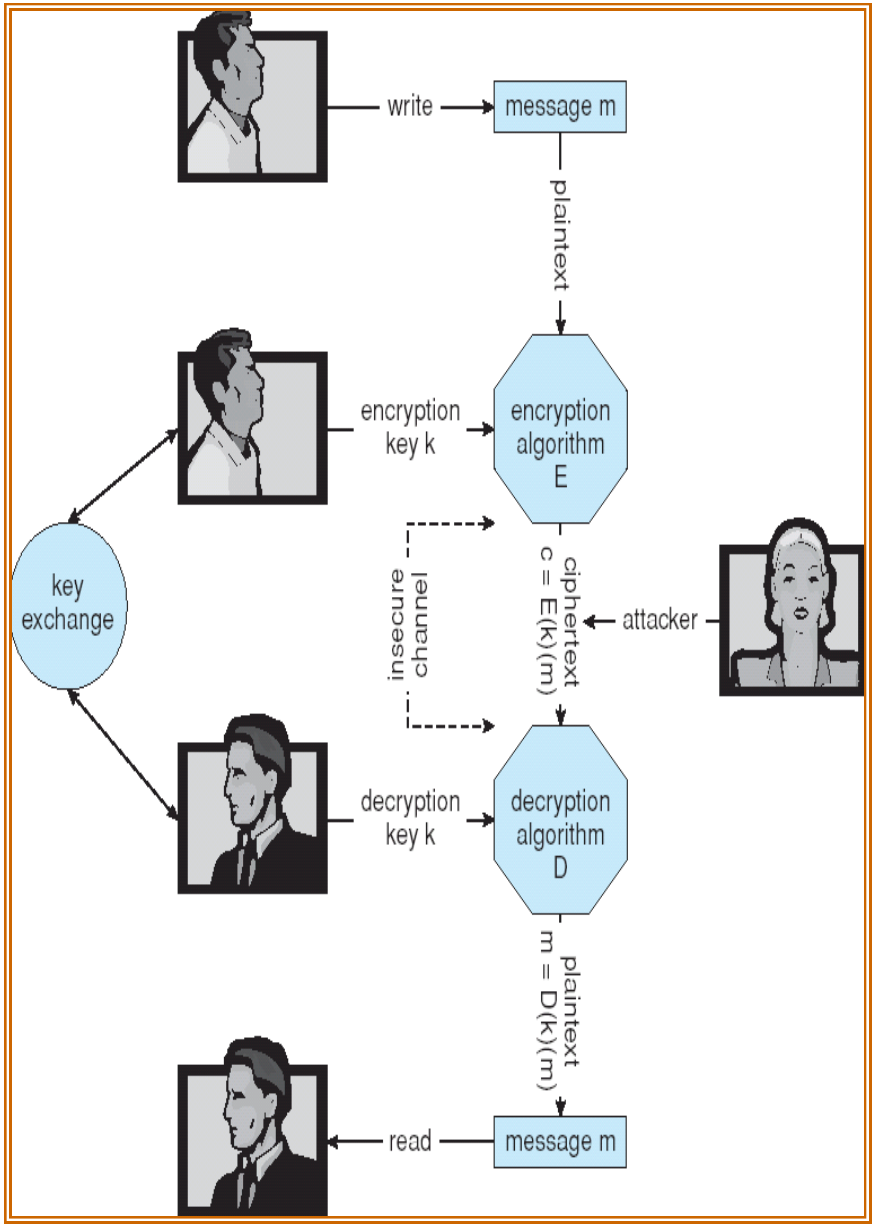
Decryption:

M = D(C, Kd)

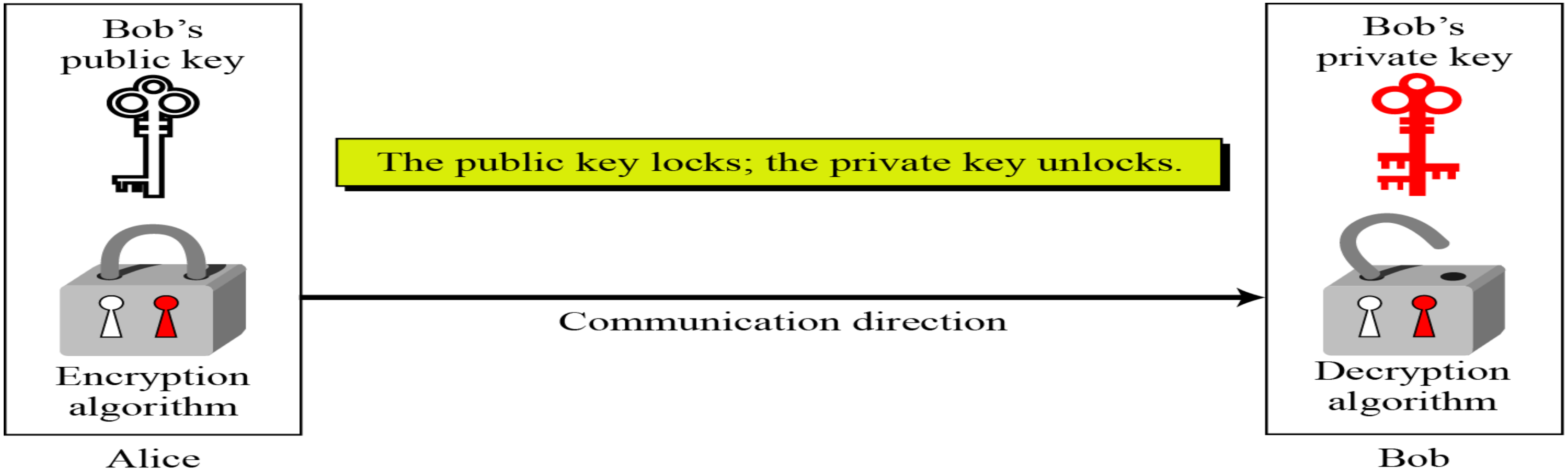
D = Decryption Algorithm

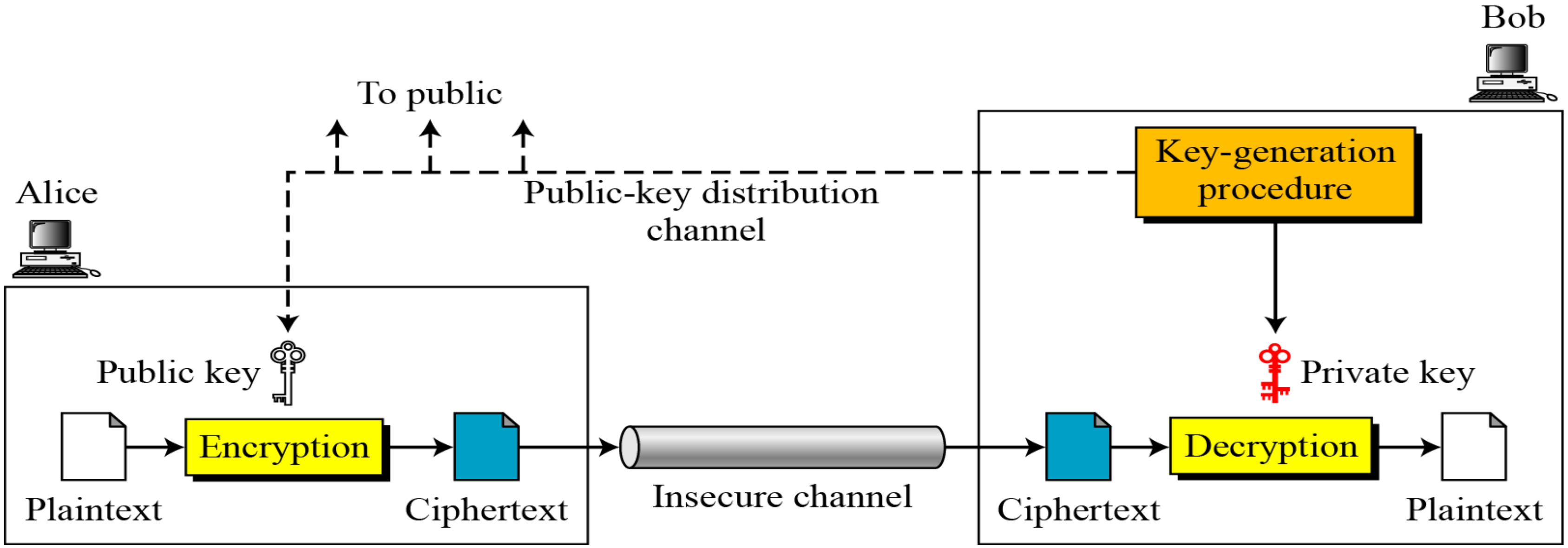
Kd = Decryption key

- Properties of good encryption technique:
- Encryption scheme depends not on the secrecy of the algorithm but on a parameter of the algorithm called the encryption key.
- Extremely difficult for an intruder to determine the encryption key.



Asymmetric Encryption

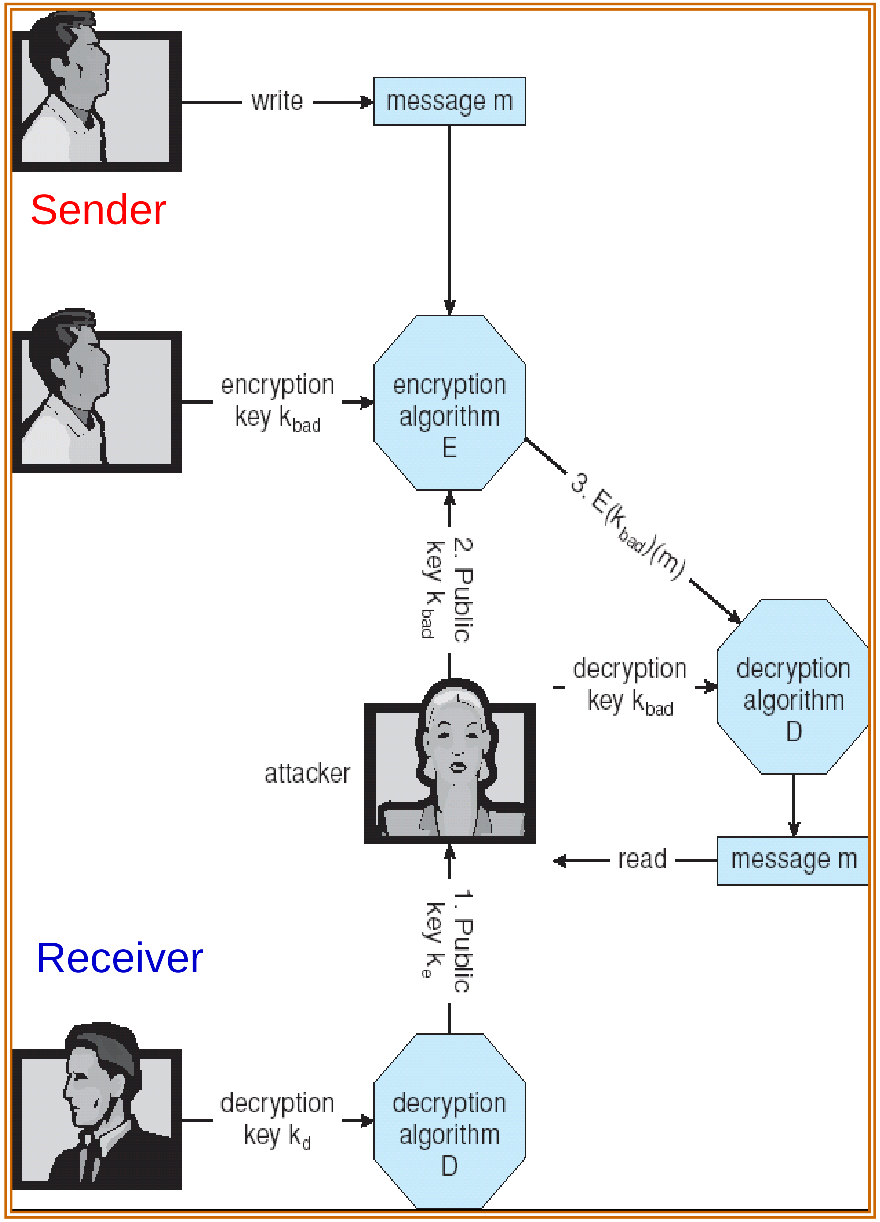




General idea of asymmetric-key cryptosystem

Man-in-the-middle Attack on Asymmetric Encryption

- Man-in-the-middle Attack on Asymmetric Cryptography.
- Here are the attack steps for this scenario:
  1. Sender (S) wishes to send a message to Receiver (R).
  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.



Encryption Example: Secure Socket Layer (SSL)

- SSL is a cryptographic protocol that enables two computers to communicate securely.
- Used between Web servers and browsers for secure communication (i.e. credit card numbers transfer, …)

- The server is verified with a **certificate**.

- Communication between each computer uses symmetric key cryptography.

The End!!

Thank you

Any Questions?