Introduction to Computer and Network Security

- **Computer** S**ecurity** basically is the protection of computer systems and information from harm, theft, and unauthorized use. It is the process of preventing and detecting unauthorized use of your computer system. It is generic name for the collection of tools designed to protect data and to thwart hackers

- **Network security** is the practice of securing a computer network from intruders, whether targeted attackers or opportunistic malware. It is used to protect data during their transmission. Application security focuses on keeping software and devices free of threats. A compromised application could provide access to the data its designed to protect.
- **Information Security** refers to the processes and methodologies which are designed and implemented to protect print, electronic, or any other form of confidential, private and sensitive information or data from unauthorized access, use, misuse, disclosure, destruction, modification, or disruption.

- Information security's primary focus is the balanced protection of the confidentiality, integrity and availability of data (also known as the CIA triad) while maintaining a focus on efficient policy implementation, all without hampering organization productivity

Confidentiality: a good example is cryptography, which traditionally is used to protect secret messages. But cryptography is traditionally used to protect data, not resources. Resources are protected by limiting

information, for example by using firewalls or address translation mechanisms.

Integrity: a good example here is that of an interrupted database transaction, leaving the database in an inconsistent state. Cryptography provides mechanisms for detecting violations of integrity, but not preventing them (e.g., a digital signature can be used to determine if data has changed).

Availability: this is usually defined in terms of "quality of service," in which authorized users are expected to receive a specific level of service (stated in terms of a metric). Denial of service attacks are attempts to block availability.

Classes of Threats

Snooping: an example is passive wiretapping, where the attacker monitors communications.

Modification: an example is active wiretapping, where the attacker injects something into a communication or modifies parts of the communication. Modification is sometimes called alteration.

Spoofing: the act of disguising a communication from an unknown source as being from a known, trusted source.

Denial of service: this may not be due to an attack, but due to limits of resources. However, the effect here is critical. If you define security in terms of what users need to access, the inability to access is a security problem regardless of whether the reason is intentional (an attack) or unintentional (not an attack).

Policies and Mechanisms

Policy: may be expressed in

- natural language, which is usually imprecise but easy to understand;
- mathematics, which is usually precise but hard to understand;
- policy languages, which look like some form of programming language and try to balance precision with ease of understanding

Mechanisms: may be

- technical, in which controls in the computer enforce the policy; for example, the requirement that a user supply a password to authenticate herself before using the computer
- procedural, in which controls outside the system enforce the policy; for example, firing someone for ringing in a disk containing a game program obtained from an untrusted source

The composition problem requires checking for inconsistencies among policies. If, for example, one policy allows students and faculty access to all data, and the other allows only faculty access to all the data, then they must be resolved (e.g., partition the data so that students and faculty can access some data, and only faculty access the other data).

Goals of Security Attacks

Prevention is ideal, because then there are no successful attacks.

Detection occurs after someone violates the policy. The mechanism determines that a violation of the policy has occurred (or is underway),

and reports it. The system (or system security officer) must then respond appropriately.

Recovery means that the system continues to function correctly, possibly after a period during which it fails to function correctly. If the system functions correctly always, but possibly with degraded services, it is said to be intrusion tolerant. This is very difficult to do correctly; usually, recovery means that the attack is stopped, the system fixed (which may involve shutting down the system for some time, or making it unavailable to all users except the system security officers), and then the system resumes correct operations.

Cryptography is the study of secure communications techniques that allow only the sender and intended recipient of a message to view its contents. ... When transmitting electronic data, the most common use of cryptography is to encrypt and decrypt email and other plain-text messages.. In data and telecommunications, cryptography is necessary when communicating over any Untrusted medium, which includes just about any network, particularly the Internet.

TYPES OF CRYPTOGRAPHIC ALGORITHMS

There are several ways of classifying cryptographic algorithms. For purposes of this paper, they will be categorized based on the number of keys that are employed for encryption and decryption, and further defined by their application and use. The three types of algorithms that will be discussed are (Figure 1):

- *Secret Key Cryptography (SKC):* Uses a single key for both encryption and decryption; also called *symmetric encryption*. Primarily used for privacy and confidentiality.
- *Public Key Cryptography (PKC):* Uses one key for encryption and another for decryption; also called *asymmetric encryption*. Primarily used for authentication, non-repudiation, and key exchange.
- *Hash Functions:* Uses a mathematical transformation to irreversibly "encrypt" information, providing a digital fingerprint. Primarily used for message integrity.

Symmetric Key Encryption

Symmetric-key encryption algorithms in cryptography use a single key or the same cryptographic keys (secret key) shared between the two parties for both encrypting the plaintext and decrypting the ciphertext. The keys could be identical or there could be a simple change to go between the two keys.

It uses RSA or other public-key protocols to securely agree upon the sharing and usage of a fresh new secret key for each message.

Asymmetric Key Encryption

Asymmetric key encryption is an encryption technique using a pair of public and private keys to encrypt and decrypt plaintext and ciphertext correspondingly when communicating.

Comparatively, asymmetric key encryption takes longer time than symmetric key encryption. It is also called public-key cryptography. Here, public keys are public and published and shared widely with everyone.

However, private keys are private and are known only to the owner. Both the keys are large numbers, paired together, however, are not identical (asymmetric).

Symmetric vs Asymmetric key cryptography

The major differences between symmetric and asymmetric key encryption are as follows −

- Symmetric key encryption is an old technique. Asymmetric key encryption is a new technique.

- Asymmetric key encryption takes much time. Symmetric key encryption takes less time.
- Symmetric key encryption is called secret-key cryptography. Asymmetric key encryption is called public-key cryptography.
- Symmetric key encryption uses only one key for both encryption and decryption, whereas asymmetric key encryption uses two keys (public and private) for both, encryption and decryption.

Drawbacks

The drawbacks of using symmetric and asymmetric key encryption are as follows −

- The drawback of symmetric key encryption is that both parties should have access to the same secret key. However, asymmetric key encryption is advantageous as both parties have access to two different keys.
- Asymmetric key encryption is a public key scheme that is susceptible to a "brute-force key search attack".
- Asymmetric key encryption has the potential security vulnerability in using asymmetric keys exposing it to a "man-in-the-middle" (MITM) attack, in which public keys communication is intercepted by an intruder (MITM) and modified by him/her providing different wrong/incorrect public keys instead.

Benefits

The benefits of using symmetric and asymmetric key encryption are as follows −

- The benefit of symmetric key encryption is it prevents MITM attacks involving the use of a Public Key Infrastructure (PKI).
- Symmetric key encryption is the simplest kind of encryption.
- Asymmetric key encryption ensures malicious people do not misuse the keys using two related keys for additional security.

  Example of symmetric encryption algorithms are DES, AES, … etc.

  Example of asymmetric encryption algorithms are RSA, Diffie-Hellman, … etc.

RSA algorithm

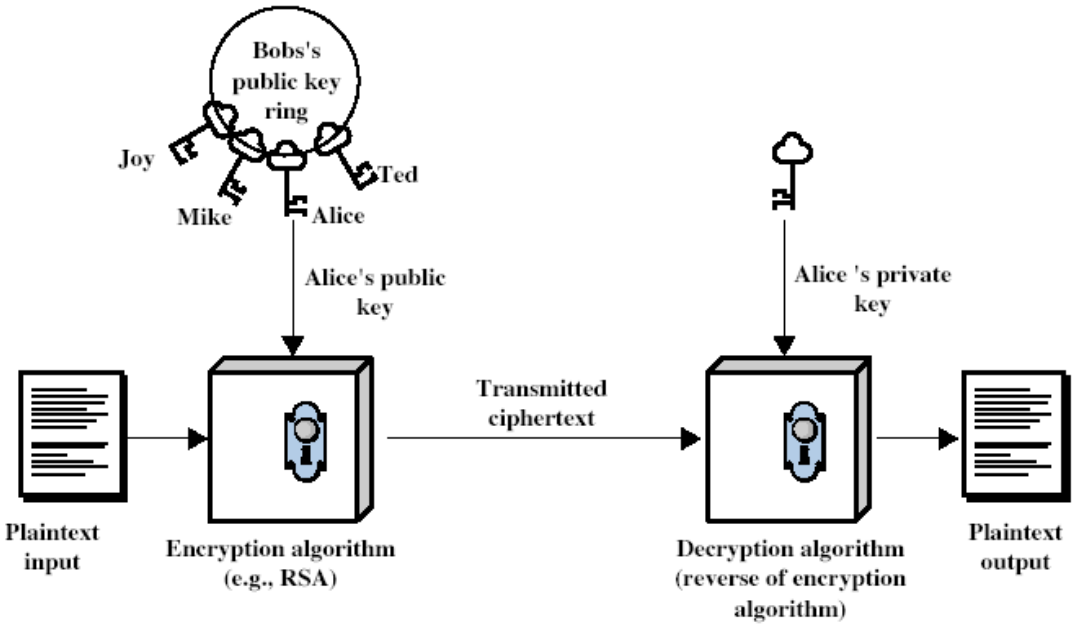
RSA (Rivest–Shamir–Adleman) is a public-key cryptosystem that is widely used for secure data transmission. It is also one of the oldest. The acronym **RSA** comes from the surnames of Ron Rivest, Adi Shamir, and Leonard Adleman, who publicly described the algorithm in 1977.

Private key cryptography

- traditional **private/secret/single key** cryptography uses **one** key
- shared by both sender and receiver
- if this key is disclosed communications are not safe any more
- **Symmetric:** hence does not protect sender from receiver forging a message & claiming is sent by sender.

Public key cryptography

- uses **two** keys – a public & a private key
- **asymmetric** since parties keys are **not** equal
- uses clever application of number theory concepts to function
- complements **rather than** replaces private key crypto



Why public key cryptography?

- developed to address two key issues:
  - The idea of public key schemes, and the first practical scheme, which was for key distribution only,
  - **digital signatures** – how to verify a message comes intact from the claimed sender

Public key characteristics

- Public-Key algorithms rely on two keys with the characteristics that it is:
  - computationally infeasible to find decryption key knowing only algorithm & encryption key
  - computationally easy to en/decrypt messages when the relevant (en/decrypt) key is known
  - either of the two related keys can be used for encryption, with the other used for decryption (in some schemes)

Confidentiality

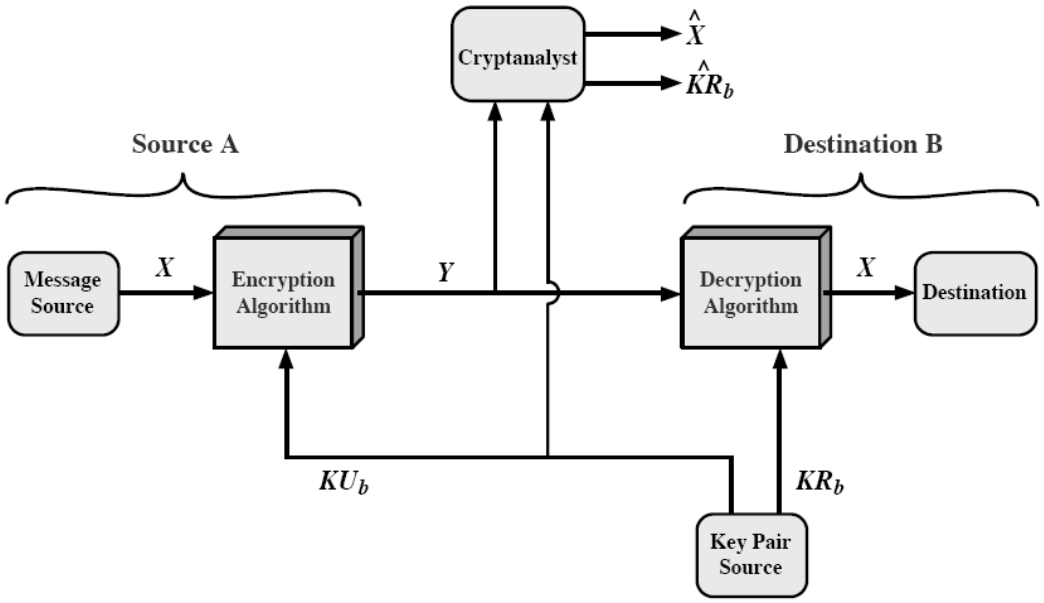


Figure 9.2 Public-Key Cryptosystem: Secrecy

Authentication

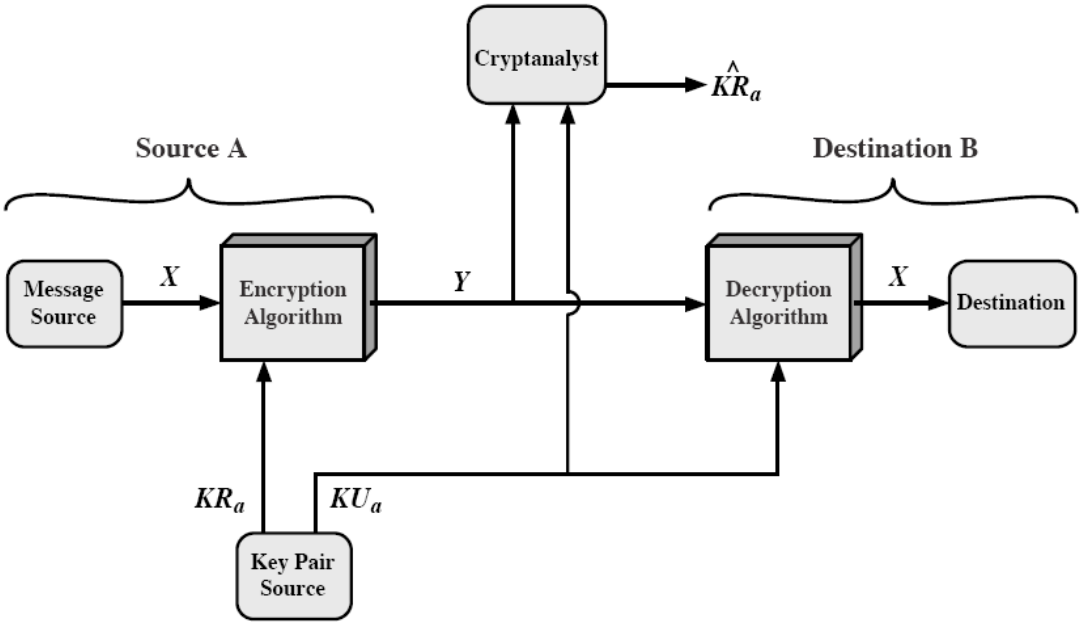


Figure 9.3 Public-Key Cryptosystem: Authentication

Public key Applications

- can classify uses into 3 categories:
  - **encryption/decryption** (provide secrecy)
  - **digital signatures** (provide authentication)
  - **key exchange** (of session keys)

Security of public key scheme

- like private key schemes brute force **exhaustive search** attack is always theoretically possible
- but keys used are too large (>512bits)
- security relies on a **large enough** difference in difficulty between **easy** (en/decrypt) and **hard** (cryptanalysis) problems
- more generally the **hard** problem is known, its just made too hard to do in practise
- requires the use of **very large numbers:**
- hence is **slow** compared to private key schemes

RSA

- RSA is the best known, and by far the most widely used general public key encryption algorithm.
- A block cipher with blocks size in the range (0 : n-1) for some n
- best known & widely used public-key scheme
- based on exponentiation in a finite (Galois) field over integers modulo a prime
- uses large integers (eg. 1024 bits)
- security due to cost of factoring large numbers

RSA Key Generation

- each user generates a public/private key pair by:
- selecting two large primes at random **p, q**
- computing their system modulus **N=p.q**
  - note **ø(N)=(p-1)(q-1)**
- selecting at random the encryption key **e**
    - **where 1<e<ø(N), gcd(e,ø(N))=1**
- solve following equation to find decryption key d
  - **e.d=1 mod ø(N) and 0≤d≤N**
  - Use the extended Euclidean algorithm to compute **d** = $\mathbf{e}^{-1}$ mod **ø(N)**
- publish their public encryption key: **KU={e,N}**

keep secret private decryption key: **KR={d,p,q}**

We need to know how to find multiplicative inverse

Finding Inverses

EXTENDED EUCLID(m, b)

1. (A1, A2, A3)=(1, 0, m);

(B1, B2, B3)=(0, 1, b)

2. if B3 = 0

return A3 = gcd(m, b); no inverse

3. if B3 = 1

return B3 = gcd(m, b); B2 = b^{-1} mod m

4. Q = A3 div B3

5. (T1, T2, T3)=(A1 – Q B1, A2 – Q B2, A3 – Q B3)

6. (A1, A2, A3)=(B1, B2, B3)

7. (B1, B2, B3)=(T1, T2, T3)

8. goto 2

Example of finding inverses

Inverse of 550 in GF(1759)

Q	A1	A2	A3	B1	B2	B3
—	1	0	1759	0	1	550
3	0	1	550	1	–3	109
5	1	–3	109	–5	16	5
21	–5	16	5	106	–339	4
1	106	–339	4	–111	355	1

RSA Use

to encrypt a message **M** the sender

- obtains **public** key of recipient **KU={e,N}**
- computes: $\mathbf{C=M^e \bmod N}$, where $\mathbf{0 \leq M < N}$

- to decrypt the ciphertext **C** the owner:
  - uses their **private** key **KR={d,p,q}**
  - computes: $\mathbf{M=C^d \bmod N}$

- note that the message **M** must be smaller than the modulus **N** (block if needed)

RSA example (encryption & Decryption)

1. Select primes: p=17 & q=11
2. Compute $n = p.q$ =17×11=187
3. Compute $ø(n)=(p–1).(q-1)$=16×10=160

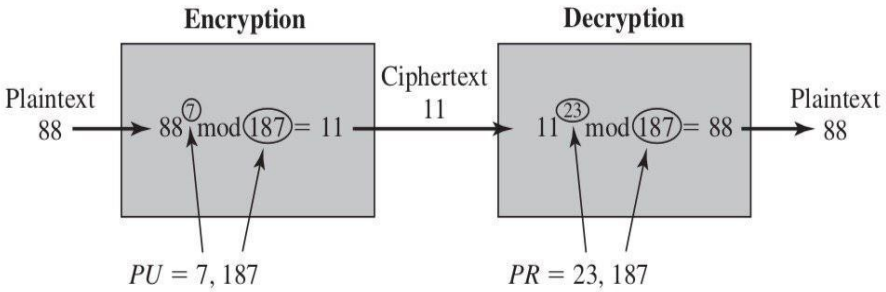
4. Select e *:* gcd(e,160)=1; choose *e*=7

5. Determine d*: de*=1 mod 160 and *d* < 160 Value is d=23

6. Publish public key KU={7,187}

7. Keep secret private key KR={23,17,11}

Assume Message =88



Example of RSA Algorithm

RSA Security: three approaches to attacking RSA:

brute force key search (infeasible given size of numbers)

2. mathematical attacks (based on difficulty of computing ø(N), by factoring modulus N):

- mathematical approach takes 3 forms:

- factor N=p.q, hence find ø(N) and then d
- determine ø(N) directly and find d
- find d directly

3.timing attacks (on running of decryption)

Diffie-Hellman Key Exchange

- first public-key type scheme proposed
- by Diffie & Hellman in 1976 along with the exposition of public key concepts
  - note: now know that Williamson (UK CESG) secretly proposed the concept in 1970
- is a practical method for public exchange of a secret key
- used in a number of commercial products

The purpose of the algorithm is to enable two users to securely exchange a key that can then be used for subsequent encryption of messages. The algorithm itself is limited to the exchange of secret values, which depends on the value of the public/private keys of the participants. The Diffie-Hellman algorithm uses exponentiation in a finite (Galois) field (modulo a prime or a polynomial), and depends for its effectiveness on the difficulty of computing discrete logarithms.

known only to the two participants

Diffie-Hellman setup

a users agree on global parameters: ➢

- large prime integer or polynomial q
- a being a primitive root mod q

- each user (eg. A) generates their key
  - chooses a secret key (number): $x_A < q$
  - compute their **public key**: $y_A = a^{xA} \bmod q$
- each user makes public that key y_A

Diffie-Hellman Example

- users Alice & Bob who wish to swap keys:
- agree on prime q=353 and a=3
- select random secret keys:
  - A chooses x_A=97, B chooses x_B=233
- compute respective public keys:
  - $y_A=3^{97} \bmod 353 = 40$ (Alice)

- $y_B=3^{233} \bmod 353 = 248$ (Bob)

➢ compute shared session key as:

- $K_{AB}= y_B{}^{xA} \bmod 353 = 248^{97} = 160$ (Alice)

- $K_{AB}= y_A{}^{xB} \bmod 353 = 40^{233} = 160$ (Bob)

Steganography

- Steganography is a means of concealing secret information within non-secret document or other media to avoid detection. It comes from the Greek words steganos, which means “covered” or “hidden,” and graph, which means “to write.” Hence, “hidden writing.”. Steganography is an additional step that can be used in conjunction with encryption in order to conceal or protect data.

- You can use steganography to hide text, video, images, or even audio data. Although the technique is old, it’s still useful enough to make us justifiably pose the question, “What is steganography in cyber security?” But before we explore its uses in today’s cyber security field, let’s get more acquainted with the overall concept by looking at some steganography examples, then wrap things up with a fun little exercise.
- Different Types of Steganography
- 1. Text Steganography − There is steganography in text files, which entails secretly storing information. In this method, the hidden data is encoded into the letter of each word.
- 2. Image Steganography − The second type of steganography is image steganography, which entails concealing data by using an image of a different object as a cover. Pixel intensities are the key to data concealment in image steganography.
- Since the computer description of an image contains multiple bits, images are frequently used as a cover source in digital steganography.
- The various terms used to describe image steganography include:
- Cover-Image - Unique picture that can conceal data.

- Message - Real data that you can mask within pictures. The message may be in the form of standard text or an image.
- Stego-Image − A stego image is an image with a hidden message.
- Stego-Key - Messages can be embedded in cover images and stego-images with the help of a key, or the messages can be derived from the photos themselves.
- 3. Audio Steganography − It is the science of hiding data in sound. Used digitally, it protects against unauthorized reproduction. Watermarking is a technique that encrypts one piece of data (the message) within another (the "carrier"). Its typical uses involve media playback, primarily audio clips.
- 4. Video Steganography − Video steganography is a method of secretly embedding data or other files within a video file on a computer. Video (a collection of still images) can function as the "carrier" in this scheme. Discrete cosine transform (DCT) is commonly used to insert values that can be used to hide the data in each image in the video, which is undetectable to the naked eye. Video steganography typically employs the following file formats: H.264, MP4, MPEG, and AVI.
- 5. Network or Protocol Steganography − It involves concealing data by using a network protocol like TCP, UDP, ICMP, IP, etc., as a cover object. Steganography can be used in the case of covert channels, which occur in the OSI layer network model.
- Steganography Examples Include Writing with invisible ink
- Embedding text in a picture (like an artist hiding their initials in a painting they've done)
- Backward masking a message in an audio file (remember those stories of evil messages recorded backward on rock and roll records?)
- Concealing information in either metadata or within a file header

- Hiding an image in a video, viewable only if the video is played at a particular frame rate
- Embedding a secret message in either the green, blue, or red channels of an RRB image
- Steganography can be used both for constructive and destructive purposes. For example, education and business institutions, intelligence agencies, the military, and certified ethical hackers use steganography to embed confidential messages and information in plain sight.
- On the other hand, criminal hackers use steganography to corrupt data files or hide malware in otherwise innocent documents. For example, attackers can use BASH and PowerShell scripts to launch automated attacks, embedding scripts in Word and Excel documents. When a poor, unsuspecting user clocks one of those documents open, they activate the secret, hidden script, and chaos ensues. This process is a favored ransomware delivery method.

Computer Viruses

A computer virus is a computer program that can copy itself and infect a computer without permission or knowledge of the user. However, the term "virus" is commonly used, albeit erroneously, to refer to many different types of malware programs. The original virus may modify the copies, or the copies may modify themselves, as occurs in a metamorphic virus. A virus can only spread from one computer to another when its host is taken to the uninfected computer, for instance by a user sending it over a network or the Internet, or by carrying it on a removable medium such as a floppy disk, CD, or USB drive.

- Meanwhile viruses can spread to other computers by infecting files on a network file system or a file system that is accessed by another computer. Viruses are sometimes confused with computer worms and Trojan horses. A worm can spread itself to other computers without needing to be transferred as part of a host, and a Trojan horse is a file that appears harmless. Both worms and Trojans will cause harm to computers when executed.
- Most personal computers are now connected to the Internet and to local area networks, facilitating the spread of malicious code. Today's viruses may also take advantage of network services such as the World Wide Web, e-mail, Instant Messaging and file sharing systems to spread, blurring the line between viruses and worms. Furthermore, some sources use an alternative terminology in which a virus is any form of self-replicating malware.

Replication strategies

In order to replicate itself, a virus must be permitted to execute code and write to memory. For this reason, many viruses attach themselves to executable files that may be part of legitimate programs. If a user tries to start an infected program, the virus' code may be executed first. Viruses can be divided into two types, on the basis of their behavior when they are executed. Nonresident viruses immediately search for other hosts that can be infected, infect these targets, and finally transfer control to the application program they infected. Resident viruses do not search for hosts when they are started. Instead, a resident virus loads itself into memory on execution and transfers control to the host program. The virus stays active in the background and infects new hosts when those files are accessed by other programs or the operating system itself.

Nonresident viruses

Nonresident viruses can be thought of as consisting of a *finder module* and a *replication module*. The finder module is responsible for finding new files to infect. For each new executable file the finder module encounters, it calls the replication module to infect that file.

Resident viruses

Resident viruses contain a replication module that is similar to the one that is employed by nonresident viruses. However, this module is not called by a finder module. .Instead, the virus loads the replication module into memory when it is executed and ensures that this module is executed each time the operating system is called to perform a certain operation. For example, the replication module can be called each time the operating system executes a file. In this case, the virus infects every suitable program that is executed on the computer.

Resident viruses are sometimes subdivided into a category *of fast infectors* and a category of *slow infectors.* Fast infectors are designed to infect as many files as possible. For instance, a fast infector can infect every potential host file that is accessed.

This poses a special problem to anti-virus software, since a virus scanner will access every potential host file on a computer when it performs a system-wide scan. If the virus scanner fails to notice that such a virus is present in memory, the virus can "piggyback" on the virus scanner and in this way infect all files that are scanned.

Fast infectors rely on their fast infection rate to spread. The disadvantage of this method is that infecting many files may make detection more likely, because the virus may slow down a computer or perform many suspicious actions that can be noticed by anti-virus software. Slow infectors, on the other hand, are designed to infect hosts infrequently. For instance, some slow infectors only infect files when they are copied. Slow

infectors are designed to avoid detection by limiting their actions: they are less likely to slow down a computer noticeably, and will at most infrequently trigger anti-virus software that detects suspicious behavior by programs. The slow infector approach does not seem very successful, however.

Computer Worm and Trojan horse

a computer worm is a self-replicating malware computer program, which uses a computer network to send copies of itself to other nodes (computers on the network) and it may do so without any user intervention. This is due to security shortcomings on the target computer. Unlike a computer virus, it does not need to attach itself to an existing program. Worms almost always cause at least some harm to the network, even if only by consuming bandwidth, whereas viruses almost always corrupt or modify files on a targeted computer.

computer worms tunnel through your computer's memory and hard drive. A computer worm is a type of virus that replicates itself, but does not alter any files on your machine. However, worms can still cause havoc by multiplying so many times that they take up all your computer's available memory or hard disk space. If a worm consumes your memory, your computer will run very slowly and possibly even crash. If the worm affects your hard disk space, your computer will take a long time to access files and you will not be able to save or create new files until the worm has been eradicated.

The biggest danger with a worm is its capability to replicate itself on your system, so rather than your computer sending out a single worm, it could send out hundreds or thousands of copies of itself, creating a huge devastating effect. One example would be for a worm to send a copy of itself to everyone listed in your e-mail address book. Then, the worm replicates and sends itself out to everyone listed in each of the receiver's address book, and the manifest continues on down the line.

Due to the copying nature of a worm and its capability to travel across networks the end result in most cases is that the worm consumes too much system memory (or network bandwidth), causing Web servers, network servers and individual computers to stop responding. In recent worm attacks such as the much-talked-about Blaster Worm, the worm has been designed to tunnel into your system and allow malicious users to control your computer remotely

Worms are hard to detect because they are typically invisible files. They often go unnoticed until your computer begins to slow down or starts having other problems. Unlike viruses and Trojan horses, worms can replicate themselves and travel between systems without any action from the user. For these reasons, it is good to have an antivirus program installed on your system that can detect and remove worms before they have a chance to replicate or spread to other computers. Security updates such as Windows Update also patch security holes that allow worms to infect your computer. So keep your security updates and virus definitions up-to-date and you should be able to keep your computer worm-free.

What Is a Trojan horse?

The Trojan Horse, at first glance, will appear to be useful software but will do damage once installed or run on your computer. Those on the receiving end of a Trojan Horse are usually tricked into opening them because they appear to be receiving legitimate software or files from a legitimate source. When a Trojan is activated on your computer, the results can vary. Some Trojans are designed to be more annoying than malicious (like changing your desktop, adding silly active desktop icons) or they can cause serious damage by deleting files and destroying information on your system. Trojans are also known to create a backdoor

on your computer that gives malicious users access to your system, possibly allowing confidential or personal information to be compromised. Unlike viruses and worms, Trojans do not reproduce by infecting other files nor do they self-replicate.

Antivirus software

Antivirus software is a term used to describe a computer program that attempts to identify, neutralize or eliminate malicious software. This type of software is so named because the earliest examples were designed exclusively to combat computer viruses; however most modern antivirus software is now designed to combat a wide range of threats, including worms, trojan horses and other malware.

Antivirus software typically uses two different techniques to accomplish this:

• Examining (scanning) files to look for known viruses matching definitions in a virus dictionary

• Identifying suspicious behavior from any computer program which might indicate infection. This technique is called heuristic analysis. Such analysis may include data captures, port monitoring and other methods.

Most commercial antivirus software uses both of these approaches, with an emphasis on the virus dictionary approach.

Following is the approaches that computer anti-virus can use it

1. Dictionary

In the virus dictionary approach, when the antivirus software looks at a file, it refers to a dictionary of known viruses that the authors of the antivirus software have identified. If a piece of code in the file matches any virus identified in the dictionary, then the antivirus software can take one of the following actions:

1. attempt to repair the file by removing the virus itself from the file

2. quarantine the file (such that the file remains inaccessible to other programs and its virus can no longer spread
3. delete the infected file

To achieve consistent success in the medium and long term, the virus dictionary approach requires periodic (generally online) downloads of updated virus dictionary entries. users can send their infected files to the authors of antivirus software, who then include information about the new viruses in their dictionaries.

Dictionary-based antivirus software typically examines files when the computer's operating system creates, opens, closes or e-mails them. In this way it can detect a known virus immediately upon receipt. Note too that a System Administrator can typically schedule the antivirus software to examine (scan) all files on the computer's hard disk .

Although the dictionary approach can effectively contain virus outbreaks in the right circumstances, virus authors have tried to stay a step ahead of such software by writing ", "polymorphic" and more recently "metamorphic" viruses, which encrypt parts of themselves or otherwise modify themselves as a method of disguise, so as to not match the virus's signature in the dictionary.

2. Suspicious behavior

The suspicious behavior approach, by contrast, doesn't attempt to identify known viruses, but instead monitors the behavior of all programs.

If one program tries to write data to an executable program, for example, the antivirus software can flag this suspicious behavior, alert a user and ask what to do.

Unlike the dictionary approach, the suspicious behavior approach therefore provides protection against brand-new viruses that do not yet exist in any virus dictionaries. However, it can also sound a large number of false positives, and users probably become desensitized to all the warnings. If the user clicks "Accept" on every such warning, then the antivirus software obviously gives no benefit to that user. This problem has worsened since, since many more non -malicious program designs came to modify other .exe files without regard to this false positive issue. Thus, most modern antivirus software uses this technique less and less.

3. Other approaches

Some antivirus software use other types of heuristic analysis. For example, it could try to emulate the beginning of the code of each new executable that the system invokes before transferring control to that executable. If the program seems to use self-modifying code or otherwise appears as a virus (if it immediately tries to find other executables, for example), one could assume that a virus has infected the executable. However, this method could result in a lot of false positives.

Yet another detection method involves using a sandbox. A sandbox emulates the operating system and runs the executable in this simulation. After the program has terminated, software analyzes the sandbox for any changes which might indicate a virus. Because of performance issues, this

type of detection normally only takes place during on-demand scans. Also this method may fail as viruses can be nondeterministic and result in different actions or no actions at all done when run - so it will be impossible to detect it from one run.