PADMASHREE KRUTARTHA ACHARYA INSTITUTE OF ENGINEERING & TECHNOLOGY, BARGARH

Cryptography And Network Security

6th SEMESTER

Computer Science and Engineering

Network Security & Cryptography

Now a day almost all It related jobs use the internet as the backbone service. Therefore it is highly essential for an IT professional to have a fare idea on the security aspect of internet service. This paper aims to provide the student with the various security threats in internet and discuss the different techniques to implement this. One of such technique is implementation of cryptography in the confidential data to be floated in the internet.

COURSE CONTENT

1. **Possible attacks on computers**
   1.1 The need for security
   1.2 Security approach
   1.3 Principles of security
   1.4 Types of attacks

2. **Cryptography concepts**

   2.1 Plain text & Cipher Text
   2.2 Substitution techniques
   2.3 Transposition techniques
   2.4 Encryption & Decryption
   2.5 Symmetric & Asymmetric key cryptography

3. **Symmetric & Asymmetric key algorithms**
   3.1 Symmetric key algorithm types
   3.2 Overview of Symmetric key cryptography
   3.3 Data encryption standards
   3.4 Over view of Asymmetric key cryptography
   3.5 The RSA algorithm
   3.6 Symmetric & Asymmetric key cryptography
   3.7 Digital signature

4. **Digital certificate & Public key infrastructure**
   4.1 Digital certificates
   4.2 Private key management
   4.3 PKIX Model
   4.4 Public key cryptography standards

5. **Internet security protocols**
   5.1 Basic concept
   5.2 Secure socket layer
   5.3 Transport layer security
   5.4 Secure Hyper text transfer protocol(SHTTP)
   5.5 Time stamping protocol (TSP)
   5.6 Secure electronic transaction (SET)

6. **User authentication**
   6.1 Authentication basics
   6.2 Password
   6.3 Authentication Tokens
   6.4 Certificate based authentication
   6.5 Biometric authentication

7. Network Security & VPN

7.1 Brief introduction of TCP/IP
7.2 Firewall
7.3 IP Security
7.4 Virtual Private Network (VPN)

Books :

1. Cryptography & Network security ; By: A. Kahate : TMH
2. Cryptography & Information security;   Pachghare   ;PHI
3. Cryptography & Network Security – Principals and Practices; By: W.Stallings, Prentice Hall.

POSSIBLE ATTACKS ON COMPUTERS

Contents.
1.1 The need for security
1.2 Security approach
1.3 Principles of security
1.4 Types of attacks

1.1 THE NEED FOR SECURITY

The computer security specifically relates to network security.

It is the security against attackers and hackers.

Network Security includes two basic securities.

- The first is the security of data information i.e. to protect the information from Unauthorized access and loss. Here network security not only means security in a single network rather in any network or network of networks. On internet or any network of an organization, thousands of important information is exchanged daily. This information can be misused by attackers. So there is a need for information security

- Second is computer security i.e. to protect your computer system from unwanted damages caused due to network. One of the major reason for such damages are the viruses and spywares that can wipe off all the information from your hard disk or sometimes they may be enough destructive and may cause hardware problems too. So there is a need to protect data and to thwart(prevent) hackers.

The security is needed for the following given reasons.

1. To protect the secret information users on the net only. No other person should see or access it.
2. To protect the information from unwanted editing, accidently or intentionally by unauthorized users.
3. To protect the information from loss and make it to be delivered to its destination properly.
4. To manage for acknowledgement of message received by any node in order to protect from denial by sender in specific situations. For example let a customer orders to purchase a few shares XYZ to the broader and denies for the order after two days as the rates go down.
5. To restrict a user to send some message to another user with name of a third one. For example a user X for his own interest makes a message containing some favorable instructions and sends it to user Y in such a manner that Y accepts the message as coming from Z, the manager of the organization.
6. To protect the message from unwanted delay in the transmission lines/route in order to deliver it to required destination in time, in case of urgency.

1.2 SECURITY APPROACH

Trusted system

A **trusted system** is a system that is relied upon to a specified extent to enforce a specified security policy. As such, a trusted system is one whose failure may break a specified security policy.

- Trusted systems are used for the processing, storage and retrieval of sensitive or classified information.
- Trusted system often use the term "reference monitor", which is an entity that occupies the logical heart of the computer system and is responsible for all access control

  decisions. Ideally, the reference monitor is
- tamperproof,
- always invoked,
- small enough to be subject to independent testing, the completeness of which can be assured.

Security model

An organization can take several approaches to implements its security model.

No security	this approach could be a decision to implement no security at all.
Security through obscurity	In this approach a system is secure simply because nobody knows about its existence and contents.
Host Security	In this approach the security for each host is enforced individually
Network Security	Host security is tough to achieve as organization grows and become more diverse. In this technique, the focus is to control network access to various hosts and their services, rather than individual host security. This is more efficient and scalable model.

Security Management Practices

Good security management practices always talk of a security policy being in place. Putting a security policy in place is actually quite tough. A good security policy and its proper implementation go a long way in ensuring adequate security management practices. A good security policy generally takes care of four key aspects, as follows

- Affordability   : -   Cost and effort in security implementation.
- Functionality   : -   Mechanism of providing security.
- Cultural issues: - Whether the policy gels well with people"s expectations, working style and beliefs.
- Legality   : - whether the policy meets the legal requirements.

Once a security policy is in place, the following points should be ensured.
- Explanation of policy to all concerned.
- Outline everybody responsibility.
- Use simple language in all communication.
- Establishment of accountability.
- Provision for exceptions and periodic reviews.

1.3 PRINCIPLES OF SECURITY

There are the four chief principles of security.

- **Confidentiality**
- **Integrity**
- **Authentication**
- **Non-repudiation**. There are two more
- **access control**
- **availability**

Let us assume that a person A wants to send a check worth $100 to another person B.

- A will like to ensure that no one except B gets the envelope and even if someone else gets it ,she does not come to know about the details of the check . This is principle of **confidentiality.**

- A and B will further like to be assured that the no one can tamper with the contents of the check (such as its amounts, date signature, name of the payee,etc.). This is the principle of **integrity.**

- B would like to be assured that the check has indeed come from A and not from someone else posing as A (as it could be a fake check in that case).this is the principle of **authentication.**

- What will happen tomorrow if B deposit the check in her account ,the money is transferred from A"s account to B"s account and then A refuses having written /sent the check ?The court of law will use A"s signature to disallow A to refute this claim and settle the dispute.
This is the principle of **non-repudiation.**

Confidentiality

- The principle of confidentiality specifies that only the sender and the intended recipients should be able to access the contents of a message.
- When we talk about confidentiality of information, we are talking about protecting the information from disclosure to unauthorized parties.
- Information has value, especially in today"s world. Bank account statements, personal information, credit card numbers, trade secrets, government documents.
- Everyone has information they wish to keep a secret. Protecting such information is a very major part of information security.
- Example of compromising confidentiality is if user of computer A sends message to user of computer B anther user C get access to this message which is not desired .
- This type of attack is called interception.

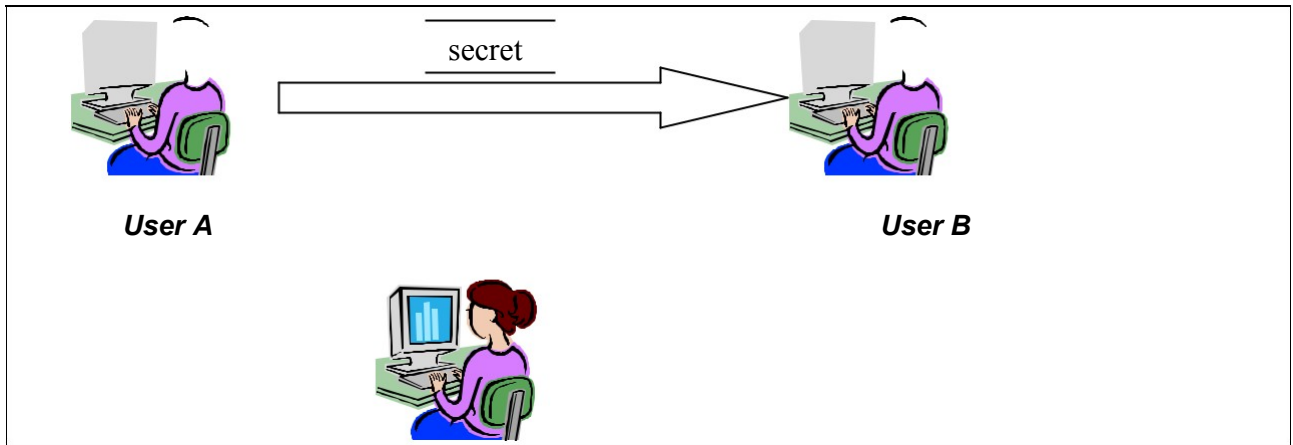*Interception causes loss of message confidentiality.*

Fig 1.1 Loss of Confidentiality

Integrity

- Integrity of information refers to protecting information from being modified by unauthorized parties.
- Information only has value if it is correct.
- Information that has been tampered with could prove costly.
- For example, if you were sending an online money transfer for $100, but the information was tampered in such a way that you actually sent $10,000, it could prove to be very costly .
- This type of attack is called modification.
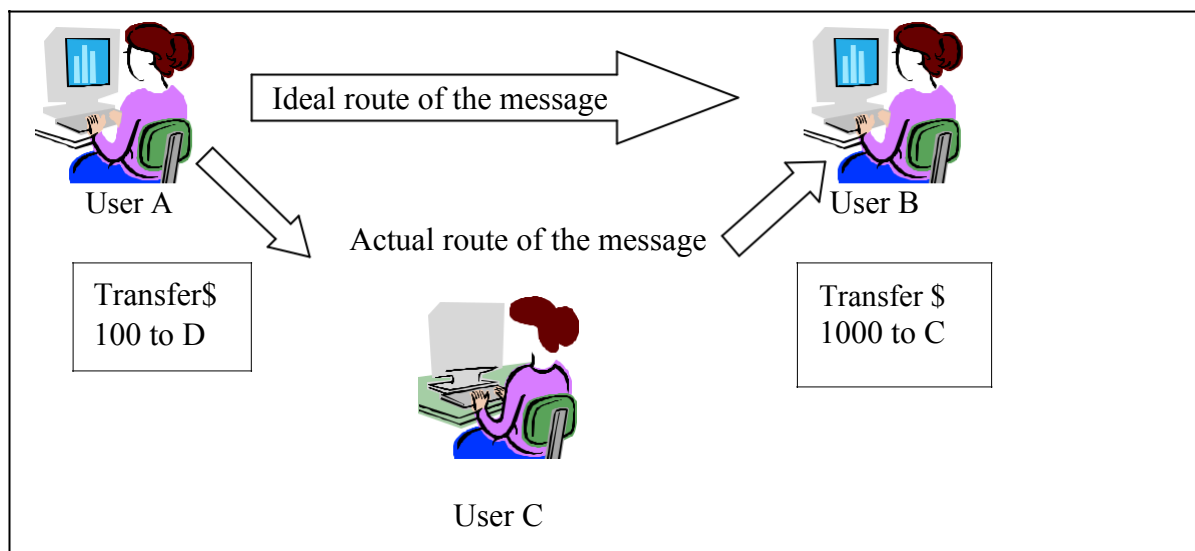
Modification causes loss of message integrity.

Fig 1.2 Loss of Integrity

Authentication

- Authentication is the process of determining the true identity of someone.
- Authentication is also used in other ways -- not just for identifying users, but also for identifying devices and data messages.

- For example suppose user C send an electronic documents to user B ,the trouble is that user C had posed as user A .
- How would user B know that the message has come from user C who is posing user A .This type of attack is called Fabrication.

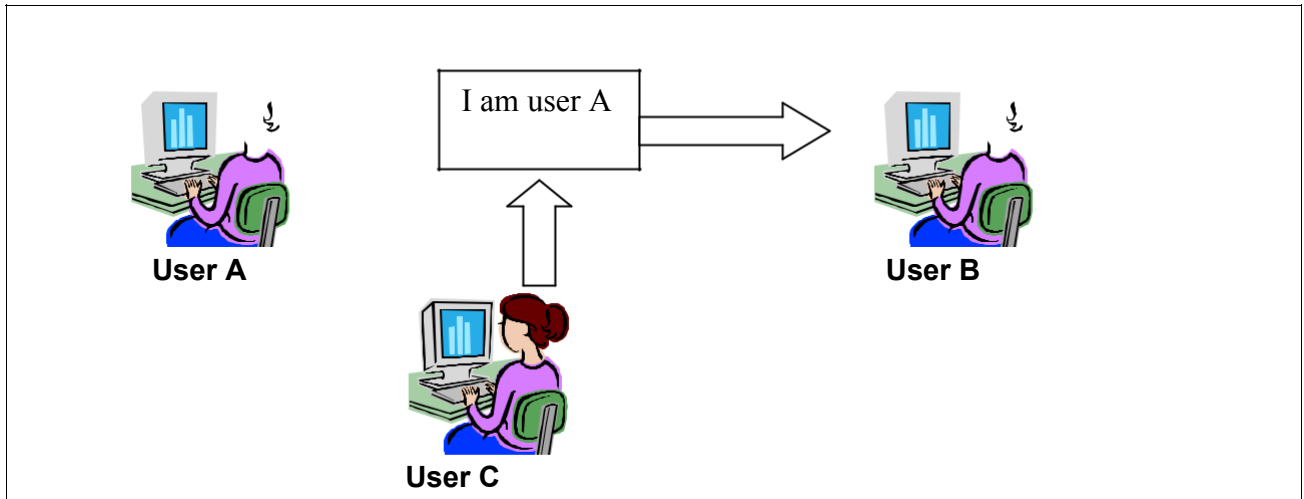*Fabrication is possible in absence of proper authentication mechanisms*

I am user A

User A

User B

User C

Fig 1.3 Absence of Authentication

Non-repudiation

- Non repudiation is the assurance that someone cannot deny something.
- Typically, non repudiation refers to the ability to ensure that a party to a contract or a communication cannot deny the authenticity of their signature on a document or the sending of a message that they originated.
- You might send registered mail, for example, so the recipient cannot deny that a letter was delivered.
- A legal document typically requires witnesses to signing so that the person who signs cannot deny having done so.

Non repudiation does not allow the sender of a message to refute the claim of not sending that message

I never sent that message which you claim have received.

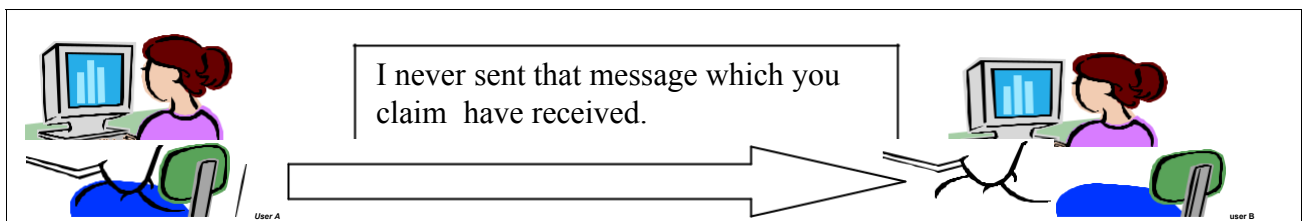User A   user B

Fig 1.4 Establishing non-reputation

Access Control

- Access control is a security technique that can be used to determine who should be able to access what.

- Physical access control limits access to campuses, buildings, rooms and physical IT assets. Logical access limits connections to computer networks, system files and data. The four main categories of access control are:
- Mandatory access control
- Discretionary access control
- Role-based access control
- Rule-based access control
- Access control systems perform authorization identification, authentication, access approval, and accountability of entities through login credentials including passwords, personal identification numbers (PINs), biometric scans, and physical or electronic keys.

Access control specifies and control who can access what.

Availability

- Availability of information refers to ensuring that authorized parties are able to access the information when needed.
- Information only has value if the right people can access it at the right times.
- Denying access to information has become a very common attack nowadays, For example, due to the intentional actions of an unauthorized user C, an authorized user A may not be able to contact a server computer B.
- This would defeat the principle of availability. Such an attack is called as interruption. ***Interruption puts the availability of resources in danger.***

1.4 TYPES OF ATTACK

General view of attacks:
For common person point of view we can classify attack into three categories.
- **Criminal attacks**:-In this attack the main aim of attackers is to maximized financial gain by attacking computer system.
- **Publicity attacks**:-Publicity attacks occur because the attackers want to see their names appear on television news channels and newspapers.
- **Legal attacks**: - This form of attack is quite novel and unique .here, The attackers tries to makes the judge or jury doubtful about the security of a computer system. This work as follows. The attacker attacks the computer system and the attacked party (say a bank or an organization) manager to take the attacker to the court. While the case is being fought, the attacker tries to convince the judge and the jury that there is inherent weakness in the computer system and that she has done nothing wrongful. The aim of the attackers is to exploit the weakness of the judge and the jury technology matters.

Technical view of attack:-

These attacks are generally classified into four categories:
1. **Interception:** - Interception causes loss of message confidentially. It means that an unauthorized party has gained access to a resource.
2. **Fabrication**:-Fabrication is possible in absence of proper authentication mechanisms. This involves creation of illegal objects on a computer system.
3. **Modification**:-Modification causes loss of message integrity.

4. **Interruption**:-Interruption cause unavailable, lost or unusable of required resource.
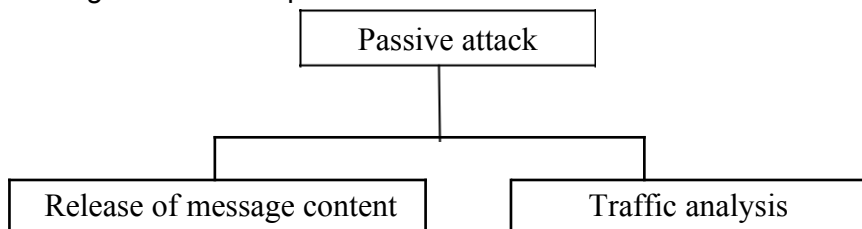
These attacks are further grouped into two types:
1.   Passive attacks
2.   Active attacks

```
                    ┌──────────────┐
                    │   Attacks    │
                    └──────┬───────┘
              ┌────────────┴────────────┐
     ┌─────────────────┐      ┌─────────────────┐
     │ Passive attacks │      │  Active attacks │
     └─────────────────┘      └─────────────────┘
```
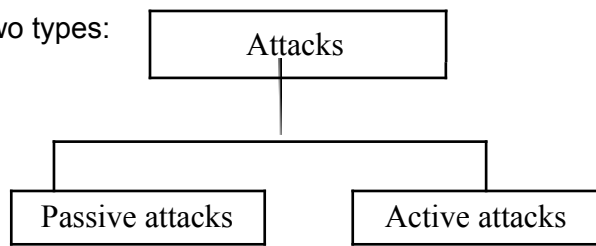
(Types of attacks)

Passive attacks:-

- In passive attack, the attacker"s goal is just to obtained information.
- This means that the attack does not modify data or harm the system.
- Passive attack do not involves any modifications to contents of an original message .Further the passive attacks are classified into two sub-categories.

```
                    ┌──────────────┐
                    │ Passive attack│
                    └──────┬───────┘
              ┌────────────┴────────────┐
  ┌─────────────────────────┐   ┌─────────────────┐
  │ Release of message content│   │ Traffic analysis│
  └─────────────────────────┘   └─────────────────┘
```

(Types of passive attacks)

Release of message content:-

- Release of message content is quite simple to understand.
- When we send a confidential email message to our friend, we desire that only she be able to access it.
- Otherwise, the content of the message are released against our wishes to someone else. Using certain security mechanisms, we can prevent release against contents.
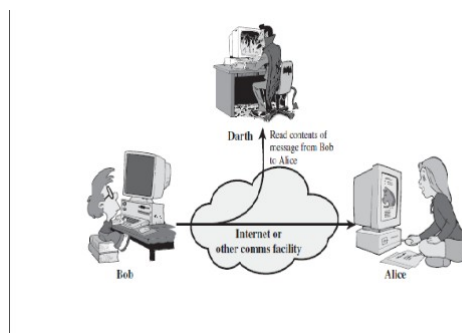

Fig 1.5 Release of Message contents

Traffic analysis:-

- If we had encryption protection, an attacker might still be able to observe the pattern of the messages
- Such attempts of analyzing messages to come up with likely pattern are known as traffic analysis attacks.

- Passive attacks are difficult to detect because they do not involves any alteration of the data.
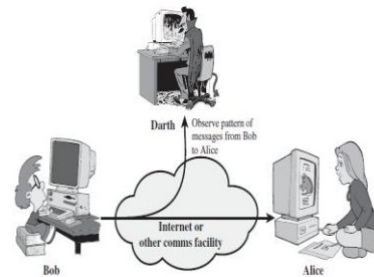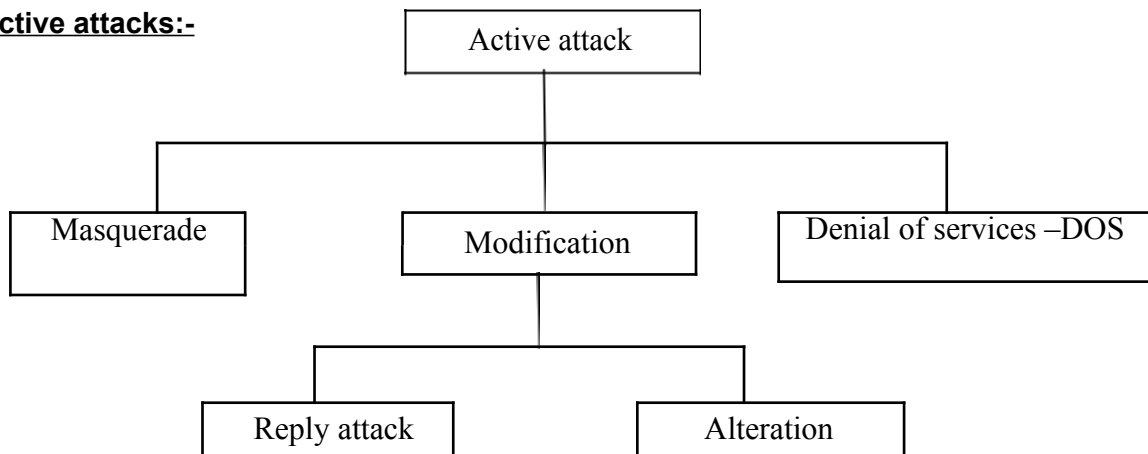

Fig 1.6 Traffic Analysis

Active attacks:-

(Types of active attacks)

☁ Active attacks may change the data or harm the system .Attacks that threatens the integrity and availability are active attacks.

☁ Active attacks are normally easier to detect than to prevent, because an attackers can launch them in a variety of ways.

☁ Active attacks are divided into three categories.

In active attack, the content of the original message are modified in some way:

- Trying to pose as another entity involves masquerade attacks.
- Modification attacks can further divided into reply attacks and alteration of messages.
- Fabrication causes denial of services (DOS) attacks.

Masquerade –

- One entity pretends to be a different entity.
- In a masquerade attack, the intruder pretends to be a particular user of a system to gain access or to gain greater privileges than they are authorized for. A masquerade may be attempted through the use of stolen login IDs and passwords, through finding security gaps in programs or through bypassing the authentication mechanism.
- Example user C might pose as user A and send a message to user B. User B might be led to believe that the message came from A.
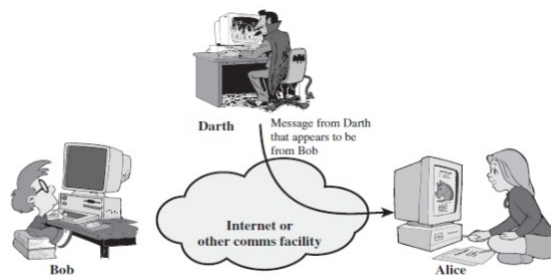
Fig 1.7 Masquerade

Replay attack –

- Involves passive capture of a data unit and its subsequent transmission to produce an unauthorized effect.
- Example suppose user A wants to transfer some amount to user C .Both users A and C have accounts with bank B. User A might send an electronic message to bank B requesting for the fund transfer. User C could capture this message and send a second copy to bank B would have no idea that this is an unauthorised message and would treated as second and different. Therefore user C would get the benefit of the found transfer twice.

Alteration of messages –

- Some portion of message is altered or the messages are delayed or recorded, to produce an unauthorized effect.
- Suppose user A(Bob) send an electronic message transfer $1000 to D"s account to bank B(alice). User C (Darth) might capture this message and change it to $10000.
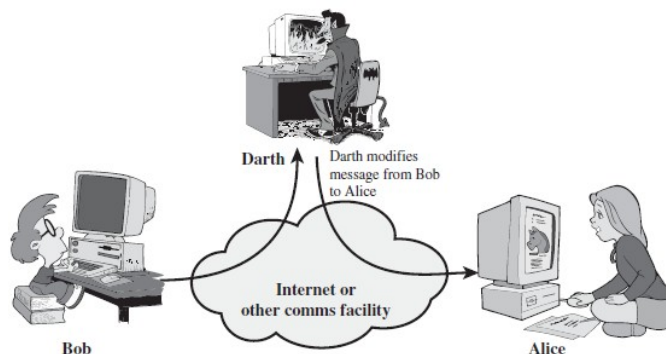
Fig 1.8 Alteration of message

Denial of service –

- Denial of service (DOS) attacks make an attempt to prevent legitimate users from accessing some services, which they are eligible for. For instance, an unauthorized user send too many login requests to a sever using random user ids one after the other in quick succession, so as to flood the network and deny other legitimate users from using the network facilities.

It is quite difficult to prevent active attacks absolutely, because to do so would require physical protection of all communication facilities and paths at all times. Instead, the goal is to detect them and to recover from any disruption or delays caused by them.

Fig 1.9 Denial of service

The practical side of Attacks:-The attacks come in a number of form in real life . They can be classified into two categories

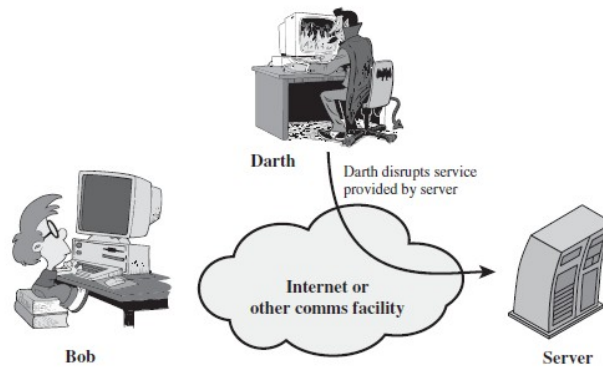1. **Application level attacks:-**These attacks happen at an application level in the sense that the attacker attempts to access modify or prevent access to information of a particular application. Example trying to obtained some ones credit card information.

2. **Network level attacks**:-These attacks generally aim at reducing the capability of network by a number of possible means. These attacks generally make an attempt to either slow down or completely bring to halt a computer network.

Computer Security (Program that attacks):- A few programs that attack computer system are:

* **Virus:-**A virus is a computer program that attaches itself to another legitimate program and causes damage of the computer system or to the network. During the life time a virus goes through four phases

1. Dormant phases:-Here the virus is idle it gate activated on certain action or event

2. Propagation phase:-In this phase a virus copy itself and each copy start creating more copies of self

3. Triggering phase:-A dormant virus move into this phase when the action event or event for which it was waiting is initiated.

4. Execution phase:-this is actual work of the virus.

   Virus can be classified into

   * Parasitic virus
   * Memory resident virus
   * Boot sector virus
   * Stealth virus
   * Polymorphic virus
   * Metamorphic virus

- Micro virus
- **Worm:**-A worm does not perform any destructive action and instead only consumes system resources to bring it down. a virus modify a program but worm does not modify a program, it replicate itself again and again.
- **Trojan horse:**-A Trojan horse allow an attacker to obtain some confidential information about a computer or network.
- **Applets and active AX:-**Java applets and active AX control are small client side programs that might cause security problems , if used by attackers with a malicious intention .

CRYPTOGRAPHY CONCEPTS

2.1 PLAIN TEXT & CIPHER TEXT

☁ **Cryptography** is the art and science of achieving security by encoding messages to make them non-readable.

☁ **Cryptanalysis** is the technique of decoding messages from a non-readable format back to readable format without knowing how they were initially converted from readable format to non-readable format.

☁ **Cryptology** is a combination of cryptography and cryptanalysis.

☁ **Plain Text :** Clear text or plain text signifies a message that can be understood by the sender, the recipient and also by anyone else who gets an access to that message.

☁ **Cipher Text :** When a plain text message is codified using any suitable scheme, the resulting message is called as cipher text.
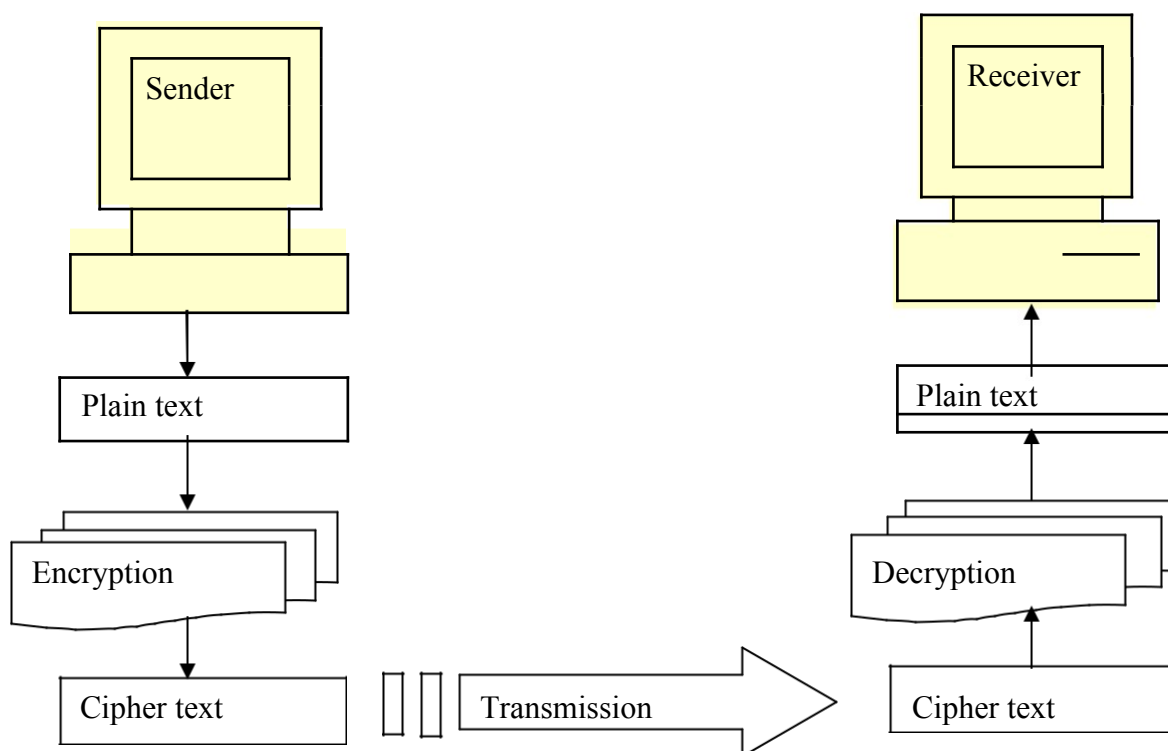
Fig 2.1 Elements of a cryptographic operation

- There are two primary ways in which a plain text message can be codified to obtain the corresponding cipher text :
  - ☁ *Substitution*
  - ☁ *Transposition*.

2.2 SUBSTITUTION TECHNIQUES

Substitution Techniques

- In the substitution cipher technique, the characters of a plain text message are replaced by other character, numbers or symbols.

Caesar cipher

- In cryptography, a Caesar cipher, also known as **Caesar's cipher**, the **shift cipher**, **Caesar's code** or **Caesar shift**, is one of the simplest and most widely known encryption techniques proposed by Julius Caesar.
- It is a type of substitution cipher in which each letter in the plaintext is replaced by an alphabet in 3 places down.
- For example, with a left shift of 3, D would be replaced by A, E would become B, and so on as shown on fig.2.2.
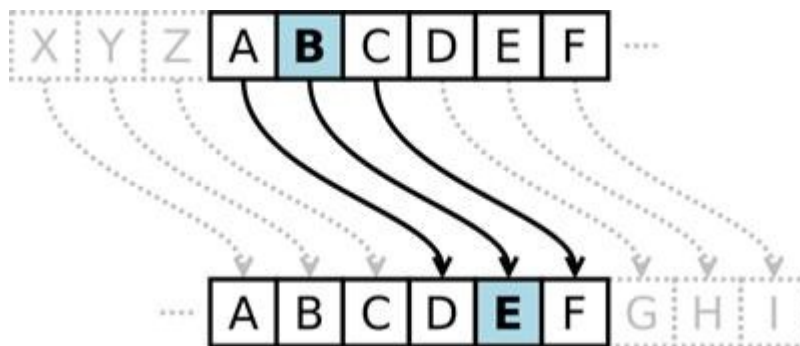
Fig 2.2 Ceasar cipher

Alphabet shifted by 3 spaces.

Fig 2.3

🚊 **Algorithm to break Caesar Cipher**

1. Read each alphabet in the cipher text message, and search for it in the second row of the replacement table (i.e. the second row of the table).
2. When a match is found, replace that alphabet in the cipher text message with the corresponding alphabet in the same column but the first row of the table (e.g. if the alphabet in cipher text is J, replace it with G).
3. Repeat the process for all alphabets in the cipher text message.

(Algorithm to break Caesar Cipher)

Modified Version of Caesar Cipher

- Modified version Caesar cipher is Caesar cipher but an alphabet A in plain text would not necessarily be replaced by D.
- It can be replaced by an any valid alphabet, i .e. by E or by F or by G and so on .
- Once replacement scheme is decided, it would be constant and will be used for all other alphabets in that message.
- The English language contains 26 alphabets thus an alphabet A can be replaced by any alphabet in the English alphabet set (i.e. b to z) of course. it does not make sense to replace an alphabet by itself(means A is replaced by A) that means each alphabet has 25 possible of replacement.

🚌 Algorithm To Break The Modified Caesar Cipher

1. Let k be a number equal to 1
2. Read the complete cipher text message.
3. Replace each alphabet in cipher text message with an alphabet that is K positions down the order.
4. Increment by 1.
5. If K is less than 26, then go to step 2. Otherwise, stop the process.
6. The original text message corresponding to the cipher text message is one of the 25 possibilities produced by the above steps

(Algorithm to break the modified Caesar cipher)

A mechanism of encoding the message so that they can send securely is called cryptography. Few terms are used in cryptography:-

Brute force attack:-

An attacks on a cipher text message, where they attacker attempt to use all possible permutation and combination is called as a brute force attack.

Cryptanalysis:-

The process of trying to break any text message to obtain the original plain text message itself is called cryptanalysis.

Cryptanalyst:-

The person attempting a cryptanalysis is called cryptanalyst.

Cipher text	K	W	U	M		P	M	Z	M
Attempt number (value of K)									
1	L	X	V	N		Q	N	A	N
2	M	Y	W	O		R	O	B	O
3	N	Z	X	P		S	P	C	P
4	O	A	Y	Q		T	Q	D	Q
5	P	B	Z	R		U	R	E	R
6	Q	C	A	S		V	S	F	S
7	R	D	B	T		W	T	G	T
8	S	E	C	U		X	U	H	U
9	T	F	D	V		Y	V	I	V
10	U	G	E	W		Z	W	J	W
11	V	H	F	X		A	X	K	X
12	W	I	G	Y		B	Y	L	Y
13	X	J	H	Z		C	Z	M	Z
14	Y	K	I	A		D	A	N	A
15	Z	L	J	B		E	B	O	B
16	A	M	K	C		F	C	P	C
17	B	N	L	D		G	D	Q	D
18	**C**	**O**	**M**	**E**		**H**	**E**	**R**	**E**
19	D	P	N	F		I	F	S	F
20	E	Q	O	G		J	G	T	G
21	F	R	P	H		K	H	U	H
22	G	S	Q	I		L	I	V	I
23	H	T	R	J		M	J	W	J
24	I	U	S	K		N	K	X	K
25	J	V	T	L		O	L	Y	L

Table 2.1 (Attempt to break modified Caesar Cipher text using all possibility)

Monalphabetic Substitution Cipher

- The major weakness of the Caesar cipher is its predictability. Once we decide to replace an alphabet the original plain text with an alphabet that is k positions up or down the order we replace all the alphabets in the plain text message with the same technique.
- Thus the cryptanalyst has to try out maximum of 25 possible attacks and she is assured to success. But in this case , each A can be replaced by alphabet (b through z),each B can also be replaced by any other random alphabet (A or C through Z) and so on.
- The crucial difference being, there is no relation between the replacement of B and replacement of A. That is ,if we have decided to replace each A with D we need not necessarily replace each B with E.

Homophonic Substitution Cipher

- The Homophonic Substitution cipher is a substitution cipher in which single plaintext letters can be replaced by any of several different cipher text letters.
- They are generally much more difficult to break than standard substitution ciphers. The number of characters each letter is replaced by is part of the key, e.g. the letter 'E' might be replaced by any of 4 different symbols(z,7,2,1), while the letter 'Q' may only be substituted by 1 symbol (k) as shown on Fig 2.4

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ

DXSFZEHCVITPGAQLKJRUOWMYBN

  9       7       3       5 0       4 6


          2


      1
```

Fig 2.4 Homophonic Substitution Cipher

Example to encipher the message DEFEND THE EAST WALL OF THE CASTLE, we find „D" in the top row, then replace it with the letter below it, „F". The second letter, „E" provides us with several choices; we could use any of „Z", „7", „2" or „1". We choose one of these at random, say „7". After continuing with this, we get the ciphertext:

Plain text: DEFEND THE EAST WALL OF THE CASTLE

Cipher text: F7EZ5F UC2 1DR6 M9PP 0E 6CZ SD4UP1

Fig 2.5 Example of Homophonic Substitution Cipher

Polygram substitution

- A simple substitution cipher substitutes for single plaintext letters. In contrast, polygram substitution ciphers involve groups of characters being substituted by other groups of characters.
- In Polygram Substitution Cipher technique replaces one block of plain text with a block of cipher text-it does not work on character by character.
- For example, HELLO can be replaced by YUQQW,but HELL could be replaced by a totally different cipher text block TEUI as shown on Fig 2.6.
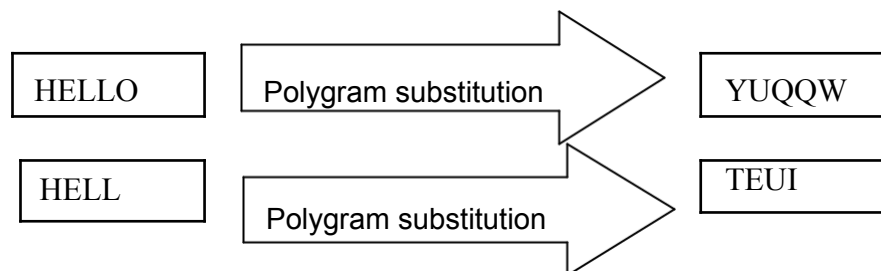

Fig 2.6 Polygram substitution Cipher

Polyalphabetic Substitution Cipher

- A *polyalphabetic substitution cipher* involves the use of two or more cipher alphabets. Instead of there being a one-to-one relationship between each letter and its substitute, there is a one-to-many relationship between each letter and its substitutes.
- The Vigenere Cipher and Beaufort Cipher are example of Polyalphabetic Substitution Cipher

- **The Vigenere Table**
  - The *Vigenere Cipher*, proposed by Blaise de Vigenere from the court of Henry III of France in the sixteenth century, is a polyalphabetic substitution based on the following table 2.2.

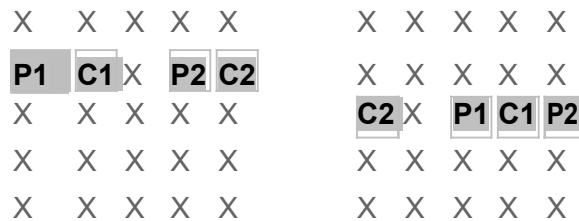|	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Table 2.2 Vigenere Table

- Each row of the table corresponds to a Caesar Cipher. The first row is a shift of 0; the second is a shift of 1; and the last is a shift of 25.
- The Vigenere cipher uses this table together with a keyword to encipher a message. For example, suppose we wish to encipher the plaintext message: "to be or not to be that is the question"
- Using the keyword RELATIONS. We begin by writing the keyword, repeated as many times as necessary, above the plaintext message. To derive the cipher text using the tableau, for each letter in the plaintext, one finds the intersection of the row given by the corresponding keyword letter and the column given by the plaintext letter itself to pick out the cipher text letter.
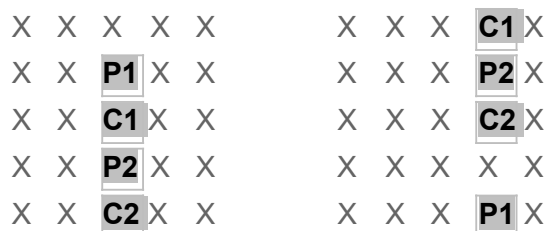
- Keyword: RELAT IONSR ELATI ONSRE LATIO NSREL
- Plaintext: TOBEO RNOTT OBETH ATIST HEQUE STION
- Ciphertext: KSMEH ZBBLK SMEMP OGAJX SEJCS FLZSY

Playfair Substitution cipher

- A Playfair cipher is a digram substitution cipher. Unlike a simple substitution cipher, which takes a message one letter at a time and replaces each letter with another letter, a Playfair cipher takes a message two letters at a time and replaces each pair of letters with another pair of letters.
- In other words, each digram is replaced with another digram. (A pair of letters is called a digram.) A given digram is always replaced by the same diagram.
- A Playfair cipher uses a keysquare containing 5 rows of 5 letters to determine the digram which should be used to replace a given digram. The key square is filled in with all the letters of the alphabet except 'J'. ('J' is left out because there is not enough room for all 26 letters and 'J' does not occur very often in normal text.)
- It is normal to use a keyword to determine the positions of the letters within the key square.
- **Using The Key Square**
- *The following three rules govern the encryption of plaintext diagrams:*
1. If the letters in the plaintext diagram are in the same row in the key square, then the letters in the cipher text diagram are immediately to the right of the plaintext letters. The first letter in the cipher text diagram is immediately to the right of the first letter in the plaintext diagram, and the second letter in the cipher text diagram is immediately to the right of the second letter in the plaintext diagram. If either plaintext letter is at the end of the row, then the corresponding cipher text letter is at the beginning.

```
X   X  X  X  X        X  X  X  X  X
P1  C1 X  P2 C2       X  X  X  X  X
X   X  X  X           C2 X  P1 C1 P2
X   X  X  X           X  X  X  X  X
X   X  X  X           X  X  X  X  X
```

2. If the letters in the plaintext digram are in the same column, then the letters in the plaintext digram are immediately below the plaintext letters. The first letter in the ciphertext digram is immediately below the first letter in the plaintext digram, and the second letter in the ciphertext digram is immediately below the second letter in the plaintext digram. If either plaintext letter is at the bottom of a column, then the corresponding ciphertext letter is the letter at the top.

```
X  X  X  X  X        X  X  X  C1 X
X  X  P1 X  X        X  X  X  P2 X
X  X  C1 X  X        X  X  X  C2 X
X  X  P2 X  X        X  X  X  X  X
X  X  C2 X  X        X  X  X  P1 X
```

3. Otherwise, the two letters in the plaintext digram are at opposite corners of a rectangle. In that case, the two letters in the cipher text digram are the letters at the remaining two corners of the rectangle. The first letter in the cipher text digram is in the same row as the first letter in the plaintext digram and the same column as the second letter in the plaintext digram, and the second letter in the cipher text digram is in the same row as the second letter in the plaintext digram and the same column as the first letter in the plaintext digram. A cipher text letter is always in the same row as its plaintext equivalent.

```
X  X  X  X  X          X   X  X  X  X
X  X  P1 X  C1          P2 X   X  C2 X
X  X  X  X  X          X  X  X  X  X
X  X  C2 X  P2          X   X  X  X  X
X  X  X  X  X          C1 X   X  P1 X
```

- **Using a keyword**
- It is common to use a keyword to determine the position of the letters within the key square. The keyword is completed by the remaining letters of the alphabet, excluding 'J', and the 25 letters placed in the key square in a pattern
- .For example, suppose that we choose the word PLAYFAIR
- We cannot put the same letter in more than one cell of the key square, so we need to remove from the keyword all repetitions. In this case, it is necessary to remove the second 'A', leavingPLAYFIR
- Now the remaining letters of the alphabet, excluding 'J', should be added to the keyword. One method is to use the letters in sequence, starting from the beginning of the alphabet:
- **PLAYFIR**BCDEGHKMNOQSTUVWXZ
- Once put in order like this, the letters can be placed in the key square. The most common method is to put them in row by row, as follows:

P L   A   Y   F

I   RSTU

VWXZB CDEGH

KMNOQ

- **Preparing the plaintext**

- A plaintext must be prepared for digram substitution.
- Firstly, the letters must be divided into pairs. Note, however, that this can cause a problem. The rules do not say what to do if the two letters in a digram are the same. Hence no plaintext diagram is allowed to contain the same letter twice. In order to avoid this problem, nulls (usually the letter 'x') have to be added to the plaintext sometimes, in order to separate identical letters.

- This has to be done whenever the two letters would otherwise fall into the same digram. Another null must be added to the end of the plaintext, if necessary, in order to complete the final diagram.
- Secondly, the letter 'J' must removed from the plaintext. It should be replaced with the letter 'I'. (This is because there is no 'J' in the keysquare. The letter 'J' is ignored because it can be replaced with the letter 'I' without causing confusion.)

- **Example**
- Suppose that we wanted to encipher the text Advance right flank to Bunker Hill, and then take up positions ready for attack.
- The first step is to divide the text into digrams:

- ad va nc er ig ht fl an kt ob un ke rh il lt he nt ak eu **px** po si ti on sr ea dy fo ra **tx** ta ck
- Note that in this case two nulls must be added. There are no 'J's to worry about. Now
- suppose that we use the following keysquare:

$$
\begin{array}{ccccc}
P & L & A & Y & F \\
I & R & S & T & U \\
V & W & X & Z & B \\
C & D & E & G & H \\
K & M & N & O & Q \\
\end{array}
$$

- According to Rule 3 above, the first digram in the plaintext, 'ad' becomes 'LE'. Likewise, the second digram, 'va' becomes 'XP', and so on. Rule 1 says that the seventh digram, 'fl' becomes 'PA' and Rule 2 says that the eighth digram, 'an' becomes 'SA'.
- This gives
- LE XP KE DS TC GU PA SA kt ob un ke rh il lt he nt ak eu px po si ti on sr ea dy fo ra tx ta ck
- When the whole plaintext is enciphered using the rules above, it becomes
- LE XP KE DS TC GU PA SA OI QZ SQ NC UD RP YR CG OS PN HS AV YK TR UR QO TS NS GL YQ SL SZ SY KP

Hill Cipher

- Invented by Lester S. Hill in 1929, the Hill cipher is a polygraphic substitution cipher based on linear algebra. Hill used matrices and matrix multiplication to mix up the plaintext. Hill's major contribution was the use of mathematics to design and analyse cryptosystems.
- This example will rely on some linear algebra and some number theory. The *key* for a hill

cipher is a matrix e.g
$$
\begin{bmatrix}
2 & 4 & 5 \\
9 & 2 & 1 \\
3 & 17 & 7 \\
\end{bmatrix}
$$

- In the above case, we have taken the size to be 3×3, however it can be any size (as long as it is square). Assume we want to encipher the message ATTACK AT DAWN. To encipher this, we need to break the message into chunks of 3. We now take the first 3

characters from our plaintext, ATT and create a vector that corresponds to the letters (replace A with 0, B with 1 ... Z with 25 etc.) to get: [0 19 19] (this is ['A' 'T' 'T']).

- To get our cipher text we perform a matrix multiplication (you may need to revise matrix multiplication if this doesn't make sense):

$$\begin{bmatrix} 2 & 4 & 5 \\ 9 & 2 & 1 \\ 3 & 17 & 7 \end{bmatrix} \begin{bmatrix} 0 \\ 19 \\ 19 \end{bmatrix} = \begin{bmatrix} 171 \\ 57 \\ 456 \end{bmatrix} \pmod{26} = \begin{bmatrix} 15 \\ 5 \\ 14 \end{bmatrix} = \text{'PFO'}$$

- This process is performed for all 3 letter blocks in the plaintext. The plaintext may have to be padded with some extra letters to make sure that there is a whole number of blocks.
- Now for the tricky part, the decryption. We need to find an inverse matrix modulo 26 to use as our 'decryption key'. i.e. we want something that will take 'PFO' back to 'ATT'. If our 3 by 3 key matrix is called K, our decryption key will be the 3 by 3 matrix K^{-1}, which is the inverse of K.

$$K^{-1} \begin{bmatrix} 15 \\ 5 \\ 14 \end{bmatrix} \pmod{26} = \begin{bmatrix} 0 \\ 19 \\ 19 \end{bmatrix} = \text{'ATT'}$$

2.3 TRANSPOSITION TECHNIQUES

- as we have seen in substitution technique we replace each plain text with new alphabets in cipher text but in transposition, we don"t not replace one alphabet with another alphabet. We just rearrange the positions of plain text and get the cipher text. Techniques used in transposition:

Rail fence Technique

- it is a type of transposition technique which rotates the position of plain text message. Example, suppose that we have a plain text message HAPPY BIRTHDAY we can convert this text in cipher text using rail fence as shown in fig 2.7.

H   p y l t   d   y

A p b r h Fig 2.7 Example of Rail fence Technique *Cipher*   a
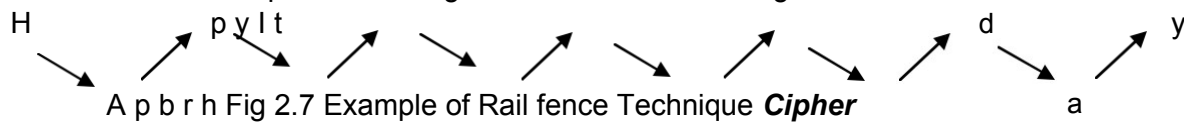
text: HPYITDYAPBRHA

🚌   *Algorithm :*

- Arrange the plain text message in sequence of diagonals as shown above .
- Read the text row by row and write it in sequence and thus we will get the cipher text.

Simple columnar transposition technique

- It rotates the position of alphabets in plain text and then find out the cipher text.

🚌   *Algorithm:*

a) write the plain text message in a rectangle of pre defined size.
b) Read the message column by column in random order of columns.

c) The message obtained by doing so is the cipher text.

- **Example,** suppose plain text that we have to encrypt is HAPPY BIRTHDAY. We can encrypt this as follows:
- Consider a rectangle with four columns and write the plain text row by row.

Col1	col2	col3	col4
H	a	p	p
Y	b	l	r
T	h	d	a
y			

Table 2.3 Simple columnar transposition technique

- Now decide the order of columns as random order. Suppose order decided is 3,1,4,2 and read the text in this order.
- Resulting text is the cipher text that is in this example cipher text is PIDHYTYPRAABH

Simple columnar transposition technique with multiple rounds

- On improve the simple columnar transposition technique, we increase the complexity of this technique by implementing the same steps twice or thrice or depending upon the security of message.

🚌 *Algorithm:*

1. write the message row by row in a rectangle of pre defined size.
2. Read the message column by column in random order of columns.
3. The message thus obtained is cipher text.
4. Repeat steps a to c as many times as needed.

- Example, consider the same PLAIN TEXT as above HAPPY BIRTHDAY.

a) Consider a rectangle with four columns and write the plain text row by row as shown in table 2.4

Col1	col2	col3	col4
H	A	P	P
Y	B	I	R
T	H	D	A

Table 2.4

b) Now decide the order of columns as random order. Suppose order decided is 3,1,4,2 and read the text in this order.

c) Resulting text is the cipher text that is in this example cipher text is PIDHYTYPRAABH d)Perform step a to c once more.

Col1	col2	col3	col4
p	i	d	h
y	t	y	p
r	a	a	b
h			

Table 2.5

d) Assume the order of column and read in that order. Suppose order is 3,1,4,2

e) Resulting text by doing so is DYAPYRHHPBITA

f) If you want iterations for more security and complexity then continue with the same steps as many times as needed.

Vernam Cipher (One-Time-Pad)

- The vernam Cipher, also called as One-Time Pad, is implemented using a random set of non-repeating characters as the input cipher text. The most significant point here is that once an input cipher text for transposition is used, it is never used again for any other message (hence the name one-time).
- The length of the input cipher text is equal to the length of the original plain text. The algorithm used in Vernam Cipher

> 1. Treat each plain text alphabet as a number in an increasing sequence, i.e. A=0, B=1, ... Z=25.
> 2. Do the same for each character of the input cipher text.
> 3. Add each number corresponding to the plain text alphabet to the corresponding input cipher text alphabet number.
> 4. If the sum thus produced is greater than 26, subtract 26 from it.
> 5. Translate each number of the sum back to the corresponding alphabet. This gives the output cipher text.

- Let us apply the Vernam Cipher algorithm to a plain text message HOW ARE YOU using a one-time pad NCBTZQARX to produce a cipher text message UQXTQUYFR.
- It should be clear that since the one-time pad is discarded after a single use, this technique is highly secure and suitable for small plain text message, but is clearly impractical for large messages. The Vernam Cipher was first implemented at AT&T with the help of a device called as Vernam Machine.
- Vernam Cipher uses a one-time pad, which is discarded after a single use and therefore, is suitable only for short messages.

Book Cipher / Running key Cipher

- The idea used in Book Cipher, also called as Running Key Cipher is quite simple and is similar in principle to the Vernam Cipher. For producing cipher text, some portion of text from a book is used, which serves the purpose of a one-time pad.
- Thus, the character from a book are used as one-time pad and they are added to the input plain text message similar to the way a one-time pad works.

2.4 ENCRYPTION & DECRYPTION

- The process of encoding plain text messages into cipher text message called as **encryption**.
- The reverse process of transforming cipher text message back to plain text is called **decryption**.
- Decryption is exactly opposite of encryption. Encryption transforms a plain text message into cipher text, where as decryption transforms a cipher text message back into plain text.
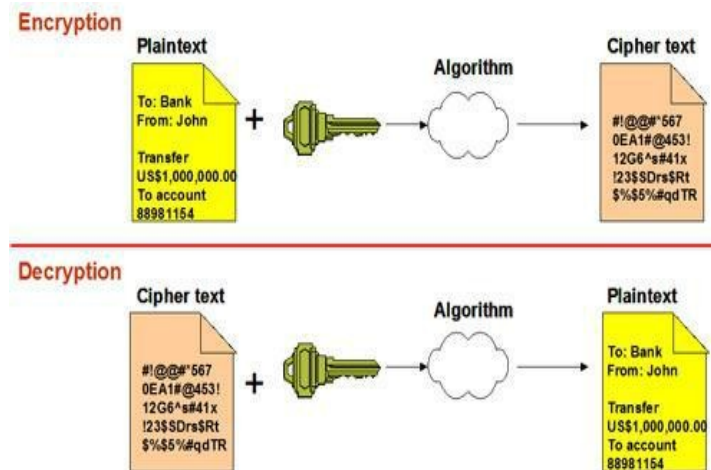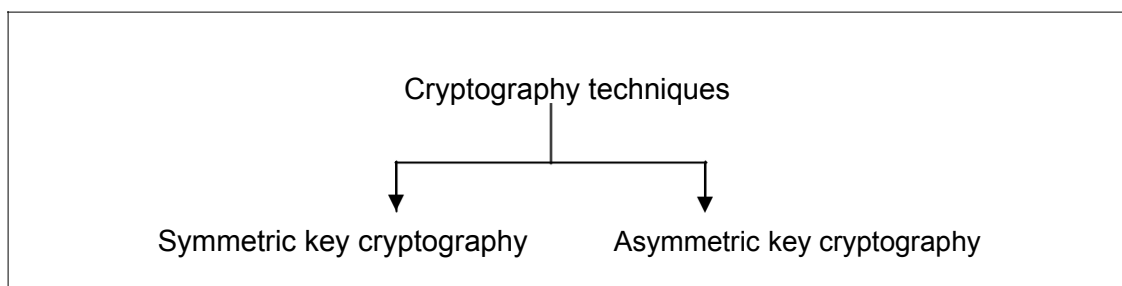
Fig 2.8 Encryption and Decryption

- The confidentiality and integrity of encrypted message is given by two factors:

- **The strength of encryption algorithm.**
- **The secrecy of the encryption key.**

- Every encryption and decryption process has two aspects the algorithm and key used for encryption and decryption**.**
- Input to encryption and decryption process is Algorithm and key

  Broadly there are two cryptography mechanisms depending on what keys are used. If the same key is used for encryption and decryption is called as **Symmetric key cryptography**. However two different key are used for decryption is called **Asymmetric key Cryptography**.

2.5 <u>SYMMETRIC & ASYMMETRIC KEY CRYPTOGRAPHY</u>

- With symmetric encryption, both parties use the same key for encryption and decryption purposes. Each user must possess the same key to send encrypted messages to each other.
- The sender uses the key to encrypt their message, and then transmits it to the receiver. The receiver, who is in procession of the same key, uses it to decrypt the message.
- The security of this encryption model relies on the end users to protect the secret key properly. If an unauthorized user were able to intercept the key, they would be able to read any encrypted messages sent by other users. It"s extremely important that the users

protect both the keys themselves, as well as any communications in which they transmit the key to another person.

- Conceptually it as similar to physical lock, perhaps a door lock. The same key is used to lock and unlock the door.
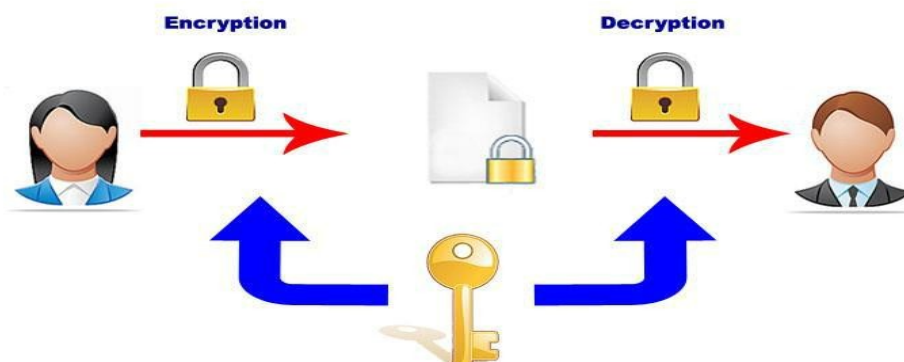

Fig 2.9 Symmetric Key Cryptography

Problem of key distribution

- Symmetric key systems have very long and strong keys but at the contrary are based on a single key for encryption and decryption with the risk of being intercepted during the key exchange between those involved in the process.
- We have the following situation:
    - When A wanted to communicate only with B, we need one lock _and _key pair(A_B).
    - When A wants to communicate with B and C ,we need two lock pair (A-B And A-C)
    - If A, B , C, D wants to communicate with each other securely, we must have 6 communicating pair A-B, A-C, A-D, B-C, B-D,AND C-D.
    - symmetric schemes require both parties to share a common secret key
    - issue is how to securely distribute this key
    - whilst protecting it from others
    - frequent key changes can be desirable
    - often secure system failure due to a break in the key distribution scheme

Diffie-Hellman Key Exchange/Agreement Algorithm

- Whitefield Diffie and Martin Hellman devised an amazing solution to the problem of key agreement or key exchange in 1976. This solution is called as the Diffie-Hellman Key Exchange/Agreement Algorithm. The beauty of this scheme is that the two parties, who want to communicate securely, can agree on a symmetric key using this technique.
- This key can then be used for encryption / decryption. However, we must note that Diffie – Hellman key exchange algorithm can be used only for key agreement, but not for encryption or decryption of messages. Once both the parties agree on the key to be used , they need to use other symmetric key encryption algorithms (we shall discuss some of those subsequently) for actual encryption or decryption of messages.
- **Description of the Algorithm**: Let us assume that Alice and Bob want to agree upon a key to be used for encrypting / decrypting messages that would be exchanged between them. Then, the Diffie-Hellman key exchange algorithm works as .
1. Firstly, Alice and Bob agree on two large prime numbers, n and g. These two integers need not be kept secret. Alice and Bob can use an insecure channel to agee on them.

2. Alice chooses another large random number x, and calculates A such that :
    $A = g^x \bmod n$
3. Alice sends the number A to Bob.
4. Bob independently chooses another large random integer y and calculates B such that :
    $B = g^y \bmod n$
5. Bob sends the number B to Alice.
6. A now computes the secret key K1 as follows :

7. B now computes the secret key K2 as follows :
    $K2 = A^y \bmod n$

- <u>Example of Algorithm</u>
    .
1. Firstly, Alice and Bob agree on two large prime numbers, n and g. These two integers need not be kept secret. Alice and Bob can use an insecure channel to agree on them.

    | Let n = 11, g=7. |

2. Alice chooses another large random number x, and calculates A such
    that : $A = g^x \bmod n$

    | Let x=3. Then, wwe have, $A=7^3 \bmod 11 = 343 \bmod 11$ |

    Let x=3. Then, wwe have, $A=7^3 \bmod 11 = 343 \bmod 11 = 2$
3. Alice sends the number A to Bob.

    | Alice sends 2 to |

4. Bob independently chooses another large random integer y and calculates B such that :
    $B = g^y \bmod n$

    | Let y = 6. Then, We have, $B = 7^6 \bmod 11 = 117649 \bmod$ |

5. Bob Sends the number B to Alice.

    | Bob sends 4 to |

6. A now computes the secret key K1 as
    follows : $K1 = B^x \bmod n$

    | We have, $K1 = 43 \bmod 11 = 64 \bmod 11$ |

7. B now computes the secret key K2 as
    follows : $K2 = A^y \bmod n$.

    | We have, $K2 = 2^6 \bmod 11 = 64 \bmod 11 = 9.$ |

Asymmetric Key Operation

- Asymmetric Encryption is a form of Encryption where keys come in pairs. What one key encrypts, only the other can decrypt. In the sense that if key A encrypts a message, and then B can decrypt it, and if key B encrypts a message, then key A can decrypt it.

- While common, this property is not essential to asymmetric encryption. Asymmetric Encryption is also known as Public Key Cryptography, since users typically create a matching key pair, and make one public while keeping the other secret.
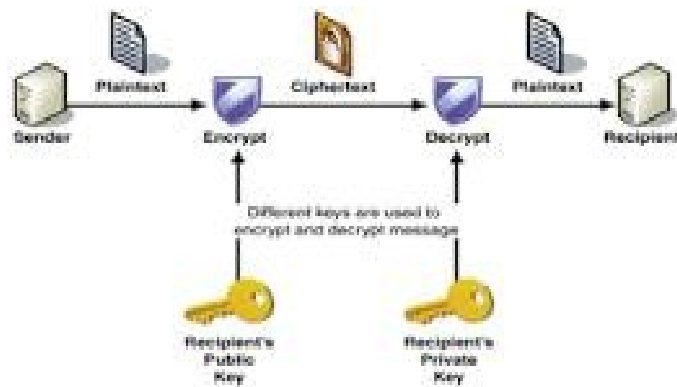
Fig 2.10 Asymmetric Key Cryptography

SYMMETRIC & ASYMMETRIC KEY ALGORITHMS

3.1 Symmetric key algorithm types
3.2 Overview of Symmetric key cryptography
3.3 Data encryption standards
3.4 Over view of Asymmetric key cryptography
3.5 The RSA algorithm
3.6 Symmetric & Asymmetric key cryptography
3.7 Digital signature

3.1 SYMMETRIC KEY ALGORITHM TYPES

- There are two aspects of algorithms: **algorithm types** and **algorithms modes.**
- **Algorithm Types:-**An algorithms type defines what size of plain text should be encrypted in each step of algorithm. Algorithms types are two types.

Algorithm Types

Stream Cipher   Block Cipher
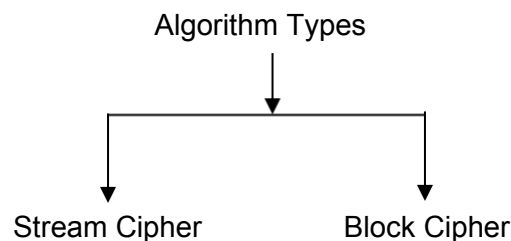
- **Stream Cipher-**In stream cipher the plain text is encrypted one byte at a time and the decryption happens one byte at a time.
- **Block Cipher-**In block cipher the plain text is encrypted one block of text at a time and decryption also takes one block at a time.

- **Algorithms Modes:-** The algorithm modes defines the details of the cryptography algorithm, once the type is decided. An algorithm mode is combination of a series of the basic algorithm steps on the block cipher and some kind of feedback from the previous steps. There are 4 types of algorithm modes.
- **Electronic Code Book-** ECB is the simplest mode of operation; the incoming plain text message is divided into blocks of 64 bits each. Each such block is then encrypted independently of the other blocks. For all blocks in a message, the same key is used for encryption.

- **Cipher Block Chaining-** CBC mode ensures that even if a block of plain text repeats in the input, these two identical plain text yields totally different cipher text blocks in the output. For this a feedback mechanism is used.

- **Cipher Feedback-**   CFB mode encrypts data in units   that"s smaller e.g. they could be of size 8 bits than a defined block size.

- **Output Feedback-**   OFB mode is extremely similar to the CFB. The only difference is that is the case of CFB, the cipher text is fed into the next stage of encryption

process. But in the case of OFB, the output of the Initial Vector (IV) encryption process is fed into the next stage of encryption process.

3.2 OVERVIEW OF SYMMETRIC KEY CRYPTOGRAPHY

- An encryption system in which the sender and receiver of a message share a single, common key that is used to encrypt and decrypt the message. Contrast this with public-key cryptology, which utilizes two keys - a public key to encrypt messages and a private key to decrypt them.

- Symmetric-key systems are simpler and faster, but their main drawback is that the two parties must somehow exchange the key in a secure way. Public-key encryption avoids this problem because the public key can be distributed in a non-secure way, and the private key is never transmitted. Symmetric-key cryptography is sometimes called *secret-key cryptography.*
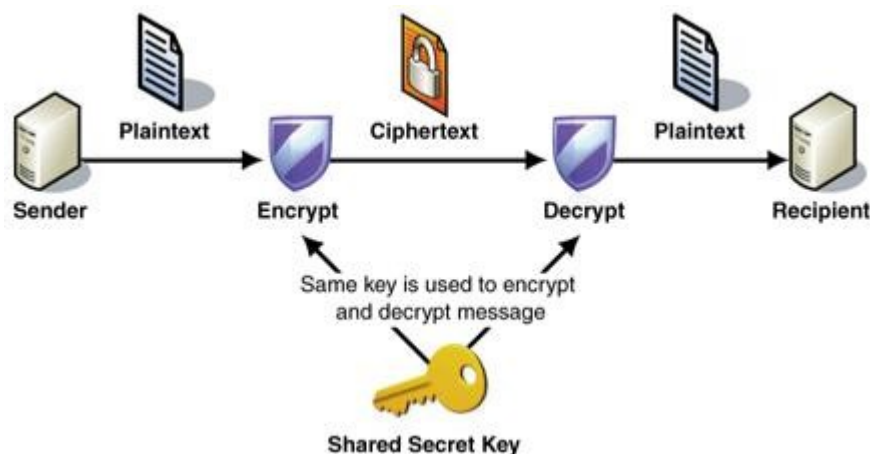

Fig 3.1 Symmetric Key cryptography

3.3 DATA ENCRYPTION STANDARDS

- The data encryption standard (DES) is a common standard for data encryption and a form of secret key cryptography (SKC), which uses only one key for encryption and decryption.
- 1972, the National Bureau of Standards (NBS) approached the Institute for Computer Sciences and Technology (ICST) to devise an encryption algorithm to secure stored and transmitted data. The algorithm would be publicly available, but its key would be top secret.
- The National Security Agency (NSA) assisted with the cryptographic algorithm evaluation processes, and in 1973, submission invitations were posted in the Federal Register. However, the submissions were unacceptable. In 1974, a second invitation was posted, which resulted in a submission from IBM. In 1975, technical specifications were published for comments in the Federal Register, and analysis and review commenced. In 1977, NBS issued the algorithm, i.e., DES.

DES WORKING PRINCIPLE

- DES is a block cipher. It encrypts data in block of size 64 bits. That is64 bits of plain text goes as the input to DES, which produce 64 bits of cipher text. The same algorithm and Key are used for encryption and decryption.

  Actually the initial key consists of 64 bits. However before DES process even starts every eight bit of the key is discarded to produce 56 key.The bit position 8,16,24,32,56,64 are discarded.

  DES is based on the two fundamental attributes Substitution (also called as confusion) And transposition (also called as Diffusion).

- **Steps of DES**

1. The 64-bit plain text block is handed over to an initial Permutation (IP) function.
2. The Initial Permutation produces 2 halves of permuted block .let Left Plain Text(LPT) and Right Plain Text(RPT).
3. Each LPT and RPT go through 16 rounds of encryption process.
4. In the end LPT and RPT are rejoined and Final Permutation (FP) is performed on the combined block.
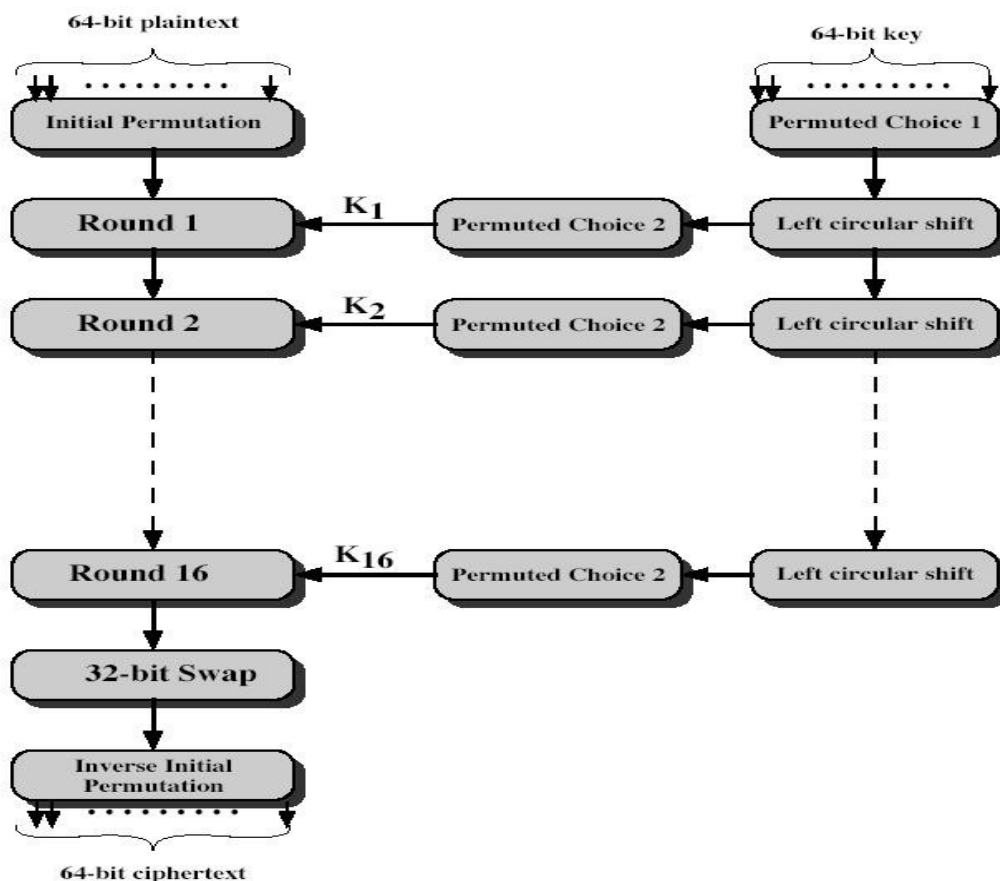5. The result of the process 64 bit cipher text.

Fig 3.2 Steps to DES

Initial Permutation

- The 64 bits of the input block to be enciphered are first subjected to the following permutation, called the initial permutation **IP**:
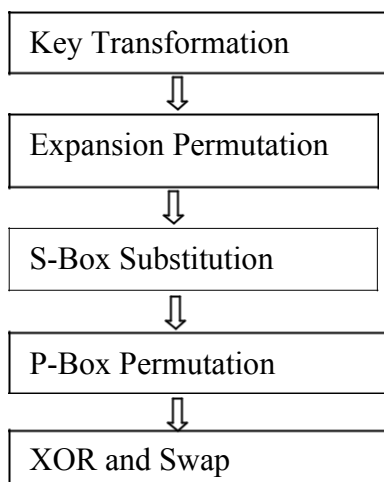
IP

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Table 3.1 Initial Permutation

- That is the permuted input has bit 58 of the input as its first bit, bit 50 as its second bit, and so on with bit 7 as its last bit.The computation which uses the permuted input block as its input to produce the preoutput block consists, but for a final interchange of blocks, of 16 iterations of a calculation that is described below in terms of the cipher function f which operates on two blocks, one of 32 bits and one of 48 bits, and produces a block of 32 bits.
- Let the 64 bits of the input block to an iteration consist of a 32 bit block **LPT** followed by a 32 bit block **RPT**. Using the notation defined in the introduction, the input block is then **LR**.

Rounds

Each of the 16 rounds in turn , consists of the following steps

Key Transformation

⇩

Expansion Permutation

⇩

S-Box Substitution

⇩

P-Box Permutation

⇩

XOR and Swap

Key Transformation

- Let **K** be a block of 48 bits chosen from the 56-bit key. A different 48 bit sub key is generated during each round using a process called as Key transformation. For this the 56-bit key is divided into 2 halves each of 28 bits. These halves are circularly shifted left

by one or two positions, depending on the round. For example, if the round number is 1, 2, 9 or 16, the shift is done by only positions. For other rounds, the circular shift is done by two positions.

Expansion Permutation

- We had two 32 bit plain text called LPT and RPT. During expansion permutation the RPT is expanded from 32 bits to 48 bits. This happens as follows
- The 32-bit RPT is divided into 8 blocks, having 4 bits in each block.
- Each 4 bit blocks of the previous step is then expanded to a corresponding 6 blocks, 2 more bits are added they are actually the repeated first and fourth of the 4 bit block.
- The Key transformation processes compress the 56 bit key to 48 bits. Then the Expansion permutation process expands the 32 bit RPT to 48 bit. Now the 48 bit key and 48 bit RPT is XORed and the resulting is given to the S-box Substitution.

S-Box Substitution

- It is the process that accepts 48-bit input from the XOR operations and produces a 32 bit output using substitution technique. The Substitution is performed by 8 substitution boxes called as S-boxes. Each of 8 boxes has a 6 –bit input and 4 bit output.
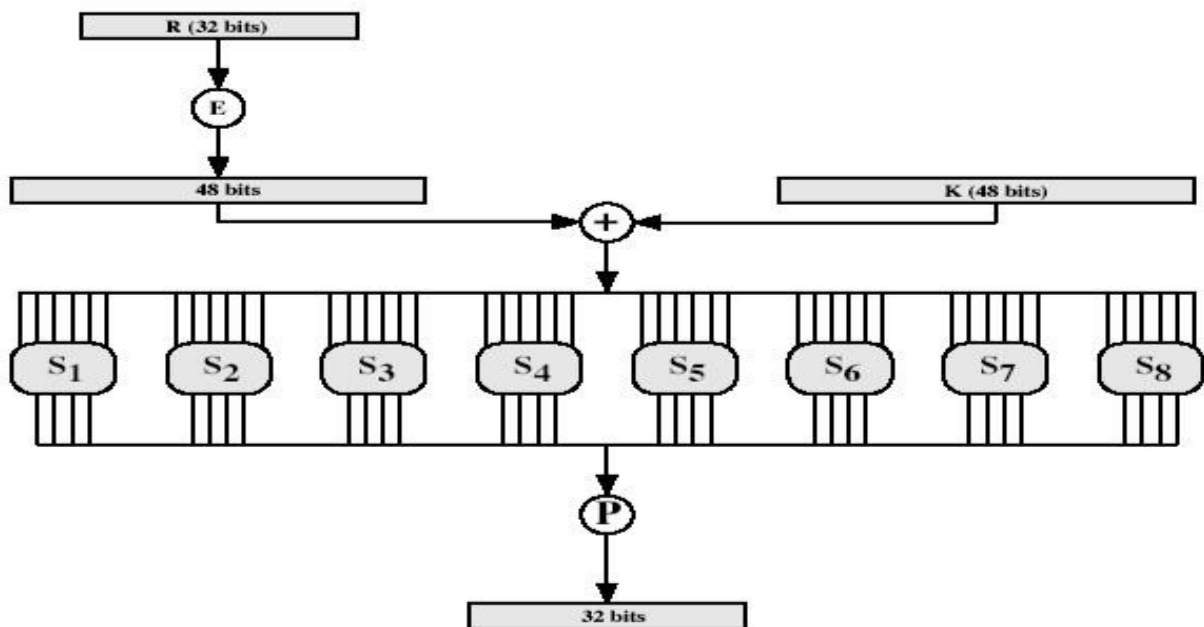

Fig 3.3 S-Box substitution

P-Box Permutation

- The out put of S-box consists of 32 bits. these 32 bits are permuted using a P-box.

XOR and Swap

- The 32 bit RPT and 32 bit LPT is XORed and swap means the LPT becomes and RPT and vice versa.

Final Purmutation

- At the end of the 16 rounds the final permutation is performed (only once).
- The nature of DES algorithm: of more concern is that cryptanalysis is possible by exploiting the characteristics of DES. The focus is the eight S-boxes used in each iteration. The design criteria for the complete algorithm has never been published and there has been speculation that the boxes were constructed in such a way that cryptanalysis is possible by an opponent who knows the weakness in the S-boxes. Although this has not been established, the US government"s "clipper project" raises many question. These are the main reasons DES is now being replaced by the AES standard .
- Using a brute-force attack by simply searching for a key is possible. However, for 56-bit key, there are 2^{56} possible key combinations, if we could search one key in 1 μs, then we need 2283 years to try all keys. (Distributed.net broke a DES-56 within 22 hours and 15 minutes, by using 100,000 PCs).

DES decryption

- The decryption process with DES is essentially the same as the encryption process andis as follows:
- Use the cipher text as the input to the DES algorithm but *use the keys* K *in reverse order*. That is, use K16 on the first iteration, K15 on the second until K1Which is used on the 16th and last iteration

Variation of DES

- In spite of its strength it is felt that with the tremendous advance in computer hard ware, DES is susceptible to possible attack. However because DES is already proven to be a very competent algorithm, it would be nice to reuse DES by making it stronger by some means, rather than writing a new cryptography algorithm. Two main variation of DES are Double DES and Triple DES.

Double DES

- Double DES is quite simple ,it does twice what DES normally does only once. Double DES uses two keys K1 and K2. The final output is encryption of encrypted text.
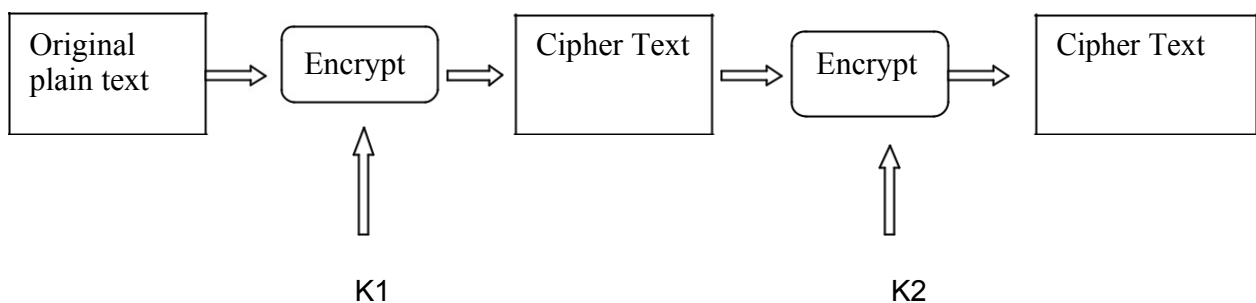

Fig 3.4 Double DES decryption

Triple DES

- Triple DES means DES three times.It comes in 2 flavour : one that uses 3 keys and second that uses 2 keys.
- Triple DES with 3 keys – the plain text encrypted with K1 , then encrypted with K2,and finally with K3.where K1 ,K2, K3 are all different from each other.
- Triple DES with 2 keys – the plain text encrypted with K1 , then encrypted with K2,and finally with K1.where K1 ,K2, are used.

3.4 OVER VIEW OF ASYMMETRIC KEY CRYPTOGRAPHY

- **Public-key cryptography**, also known as **asymmetric cryptography**, is a class of cryptographic algorithms which requires two separate keys, one of which is *secret* (or *private*) and one of which is *public*. Although different, the two parts of this key pair are mathematically linked. The public key is used to encrypt plaintext or to verify a digital signature; whereas the private key is used to decrypt cipher text or to create a digital signature. The term "asymmetric" stems from the use of different keys to perform these opposite functions, each the inverse of the other – as contrasted with conventional ("symmetric") cryptography which relies on the same key to perform both.
- Public-key algorithms are based on mathematical problems which currently admit no efficient solution that are inherent in certain integer factorization, discrete logarithm, and elliptic curve relationships. It is computationally easy for a user to generate their own public and private key-pair and to use them for encryption and decryption.
- The strength lies in the fact that it is "impossible" (computationally infeasible) for a properly generated private key to be determined from its corresponding public key. Thus the public key may be published without compromising security, whereas the private key must not be revealed to anyone not authorized to read messages or perform digital signatures. Public key algorithms, unlike symmetric key algorithms, do *not* require a secure initial exchange of one (or more) secret keys between the parties.
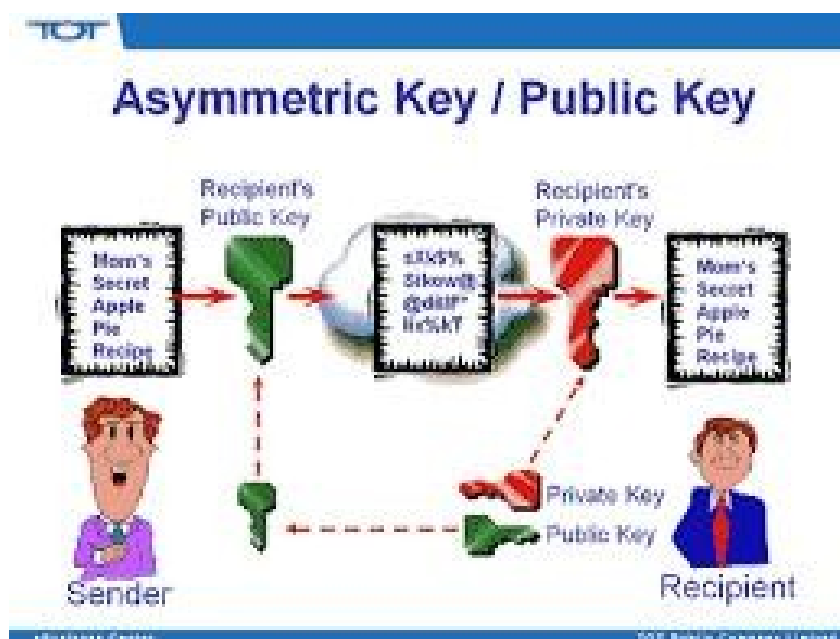

Fig 3.5 Asymmetric key/Public key Criptograpy

3.5 <u>THE RSA ALGORITHM</u>

- The RSA algorithm is named after Ron Rivest, Adi Shamir and Len Adleman, who invented it in 1977 [RIVE78]. The basic technique was first discovered in 1973 by Clifford Cocks [COCK73] of CESG (part of the British GCHQ) but this was a secret until 1997.
- The RSA cryptosystem is the most widely-used public key cryptography algorithm in the world. It can be used to encrypt a message without the need to exchange a secret key separately.
- The RSA algorithm can be used for both public key encryption and digital signatures. Its security is based on the difficulty of factoring large integers.
- Party A can send an encrypted message to party B without any prior exchange of secret keys. A just uses B's public key to encrypt the message and B decrypts it using the private key, which only he knows. RSA can also be used to sign a message, so A can sign a message using their private key and B can verify it using A's public key

<u>RSA algorithm</u>

1. Choose two large prime number P and Q .
2. Calculate N =P and Q .
3. Select the public key (i.e the encryption key) E such that it is not a factor of (P-1)and(Q-1).

4. Select the private key (i.e the decryption key)D such that the following equation is true: (D*E) mod (P-1) *(Q-1)=1
5. For encryption , calculate the cipher text CT from the plain text PT as followes: $CT=PT^E$ mod N.
6. Send CT as the cipher text to the receiver .
7. For decryption ,calculate the plain text PT from the cipher text CT as followes:$PT=CT^D$ mod N.

<u>RSA Algorithm Example</u>

- Choose p = 3 and q = 11
- Compute n = p * q = 3 * 11 = 33
- Compute $\varphi(n)$ = (p - 1) * (q - 1) = 2 * 10 = 20
- Choose e such that 1 < e < $\varphi(n)$ and e and n are coprime. Let e = 7
- Compute a value for d such that (d * e) % $\varphi(n)$ = 1. One solution is d = 3 [(3 * 7) % 20 = 1]
- Public key is (e, n) => (7, 33)
- Private key is (d, n) => (3, 33)
- The encryption of *m = 2* is *c = 2^7 % 33 = 29*
- The decryption of *c = 29* is *m = 29^3 % 33 = 2*

3.6 SYMMETRIC & ASYMMETRIC KEY CRYPTOGRAPHY

- Asymmetric key cryptography solves the problem of key agreement and key exchange. However this does not solves all the problems in a practical security infrastructure. More specifically symmetric and asymmetric key cryptography differ in certain other respects.

	Symmetric key cryptography	Asymmetric key cryptography
Key used for encryption/decryption	same key is used for encryption and decryption	One key used for encryption and another, different key is used for decryption
Speed of encryption/decryption size of resulting encrypted text	Very fast Usually same as or less than the original clear text size	Slower More than the original clear text size
Key agreement/exchange number of keys required as compared to the number of participants in the message exchange	A big problem Equal about the square of the number of participants, so scalability is an issue	No problem at all Same as the number of participants, so scales up quite well
Usage	Mainly for encryption and decryption(confidentiality),cannot be used for digital signatures(integrity and non-repudiation checks)	Can be used for encryption and decryption(confidentiality)as well as for digital signatures (integrity and non-repudiation checks)

Table 3.2 Difference between Symmetric and Asymmetric Key Criptography

The Best Of Both Worlds
- Symmetric key cryptography and asymmetric key cryptography are combined to have a very efficient security solution i.e. Digital Envelope.
- A digital envelope use to protect a digital document from being visible to anyone other than the intended recipient. The following are possible reasons for using digital envelopes:
- Sending confidential data or documents across (possibly) insecure communication
- lines Storing confidential data or documents (for example, company-internal reports)

Process Flow

- The following explains what happens at each step:

1. The message is encrypted using symmetric encryption. Typically, a newly generated random message key (secret key) is used for the encryption. Symmetric encryption means that the same key is used for both encryption and decryption (a secret key). Anyone wanting to decrypt the message needs access to this key.
2. To transfer the secret key between the parties, the secret key is encrypted using the recipient"s public key.
3. The encrypted document and the encrypted message key are packed together in a single data packet to save or send to the intended recipient
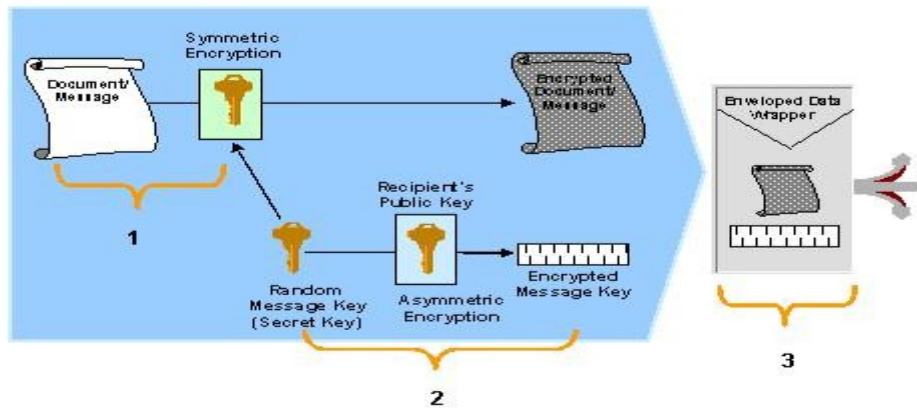
Fig 3.6 Digital Envelope

3.7 DIGITAL SIGNATURE

- A **digital signature** is basically a way to ensure that an electronic document (e-mail, spreadsheet, text file, etc.) is **authentic**. Authentic means that you know who created the document and you know that it has not been altered in any way since that person created it.

- A **digital signature** is a mathematical scheme for demonstrating the authenticity of a digital message or document. A valid digital signature gives a recipient reason to believe that the message was created by a known sender, such that the sender cannot deny having sent the message (authentication and non-repudiation) and that the message was not altered in transit (integrity). Digital signatures are commonly used for software distribution, financial transactions, and in other cases where it is important to detect forgery or tampering.
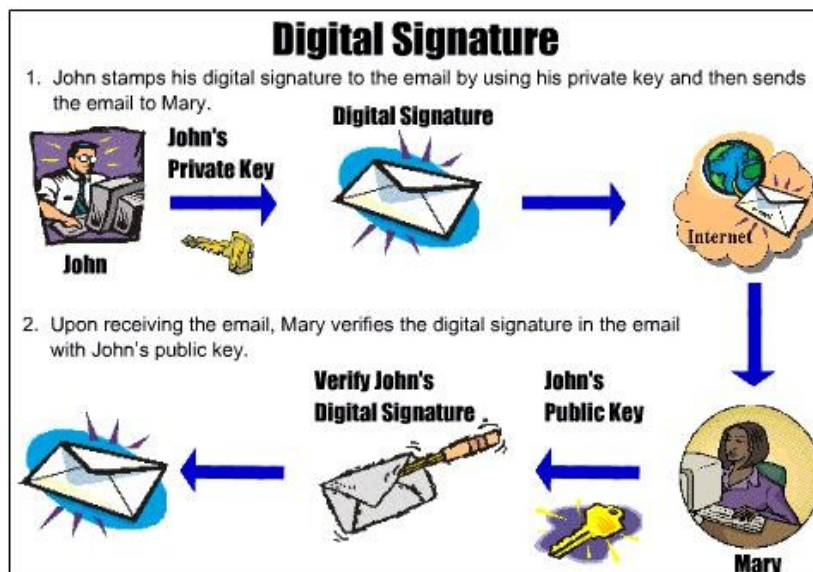

Fig 3.7 Digital Signature

<u>**Explanation**</u>

Using Bob and Alice, we can illustrate how a digital signature (standard electronic signature) is applied and verified.

Fig 3.8 Digital Signature

Getting a Private and Public Key From Bob"s perspective, the signing operation can be as simple as a click of a button. But several things happen with that one click:

1. **Step 1: Getting a Private and Public Key**

   In order to digitally sign a document, Bob needs to obtain a private and public key, which is a one-time process. The private key, as the name implies, is not shared and is used only by the signer. The public key is openly available and used by those that need to validate the signer"s digital signature.

2. **Step 2: Signing an Electronic Document**

   Initiate the signing process - Depending on the software used, Bob needs to initiate the signing process.

   Create a digital signature - A unique digital fingerprint of the document (sometimes called a message digest or document hash) is created using a mathematical algorithm (such as SHA-1). Even the slightest difference between two documents would create a separate digital fingerprint of each.

   Append the signature to the document - The hash result and the user"s digital certificate (which includes the user"s public key) are combined into a digital signature (by using the user"s private key to encrypt the document hash). The resulting signature is unique to both the document and the user. Finally, the digital signature is appended to the document. Bob sends the signed document to Alice. Alice uses Bob"s public key (which is included in the digital certificate) to authenticate Bob"s signature and to ensure that no changes were made to the document after it was signed.

3. **Step 3: Validating a Digital Signature**

   Initiate the validation process - Depending on the software used, Alice needs to initiate the validation process .Using Bob"s public key, Alice decrypts his digital signature and receives the original document (the document fingerprint).

   Compares the document fingerprint with her calculated one - Alice"s software then calculates the document hash of the received document and compares it with the original document hash (from the previous step). If they are the same, the signed document has not been altered.
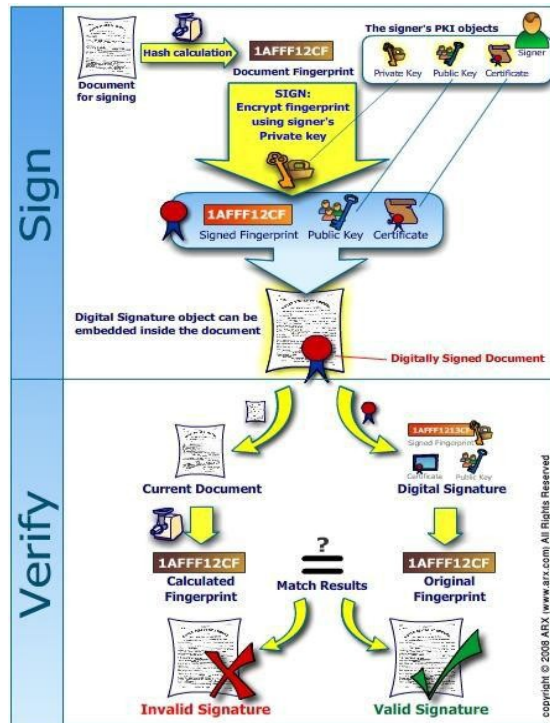
Fig 3.9 Signing and verification process of digital signature

- Signing an Electronic Document. There is yet another factor involved. How can Alice know whether Bob is indeed the same person she intends to conduct business with? Bob needs to be certified by a trusted third party that knows him and can verify that he is indeed who he claims to be. These trusted third parties are called Certificate Authorities (CA). They issue certificates to ensure the authenticity of the signer. Certificates can be compared to passports issued by countries to their citizens for world travel. When a traveler arrives at a foreign country, there is no practical way to authenticate the traveler"s identity. Instead, the immigration policy is to trust the passport issuer (in PKI terminology, this is the CA) and use the passport to authenticate its holder in the same way that Alice uses the CA"s certificate for authenticating Bob"s identity.

Message Digest Functions

- The digest is sometimes also called the "hash" or "fingerprint" of the input. Hash functions are used in many situations where a potentially long message needs to be processed and/or compared quickly. The most common application is the creation and verification of digital signatures.Message digest functions also called *hash functions* , are used to produce digital summaries of information called message digests. Message digests (also called *hashes*) are commonly 128 bits to 160 bits in length and provide a digital identifier for each digital file or document. Message digests are designed to protect the integrity of a piece of data or media to detect changes and alterations to any part of a message..
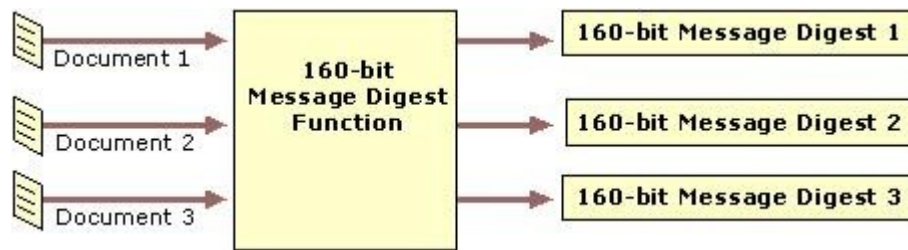
Fig 3.10 Example of the Message Digest

MD5

- MD5 was designed by well-known cryptographer Ronald Rivest in 1991. In 2004, some serious flaws were found in MD5. Message-Digest 5, known simply as MD5, is one of the quickest and simplest ways to add security to the files and messages that you send and transfer.
- It looks complicated, but it actually relies on a few simple ideas. To get a MD5 hash all you need to do is input your message string into a MD5 generator. This is a web app that will apply an algorithm to the string to create a MD5 hash.The MD5 function is a cryptographic algorithm that takes an input of arbitrary length and produces a *message digest* that is 128 bits long.
- The digest is sometimes also called the "hash" or "fingerprint" of the input. MD5 is used in many situations where a potentially long message needs to be processed and/or compared quickly. The most common application is the creation and verification of digital signatures.

How MD5 works

- **Preparing the input**
- The MD5 algorithm first divides the input in **blocks** of 512 bits each. 64 Bits are inserted at the end of the last block. These 64 bits are used to record the length of the original input. If the last block is less than 512 bits, some extra bits are 'padded' to the end. Next, each **block** is divided into 16 **words** of 32 bits each. These are denoted as M_0 ... M_{15}.
- **MD5 helper functionsThe buffer**
- MD5 uses a buffer that is made up of four **words** that are each 32 bits long. These words are called A, B, C and D. They are initialized as word A: 01 23 45 67
-
- word B: 89 ab cd ef
- word C: fe dc ba 98
- word D: 76 54 32 10
- **The table**
- MD5 further uses a table K that has 64 elements. Element number i is indicated as K_i. The table is computed beforehand to speed up the computations. The elements are computed using the mathematical sin function:

$K_i = abs(sin(i + 1)) * 2^{32}$

- **Four auxiliary functions**
- In addition MD5 uses four auxiliary functions that each take as input three 32-bit words and produce as output one 32-bit word. They apply the logical operators and, or, not and xor to the input bits.
- F(X,Y,Z) = (X and Y) or (not(X) and Z)

    G(X,Y,Z) = (X and Z) or (Y and not(Z))

    H(X,Y,Z) = X xor Y xor Z

    I(X,Y,Z) = Y xor (X or not(Z))

- **Processing the blocks**
- The contents of the four buffers (A, B, C and D) are now mixed with the words of the input, using the four auxiliary functions (F, G, H and I). There are four *rounds*, each involves 16 basic *operations*. One operation is illustrated in the figure below.
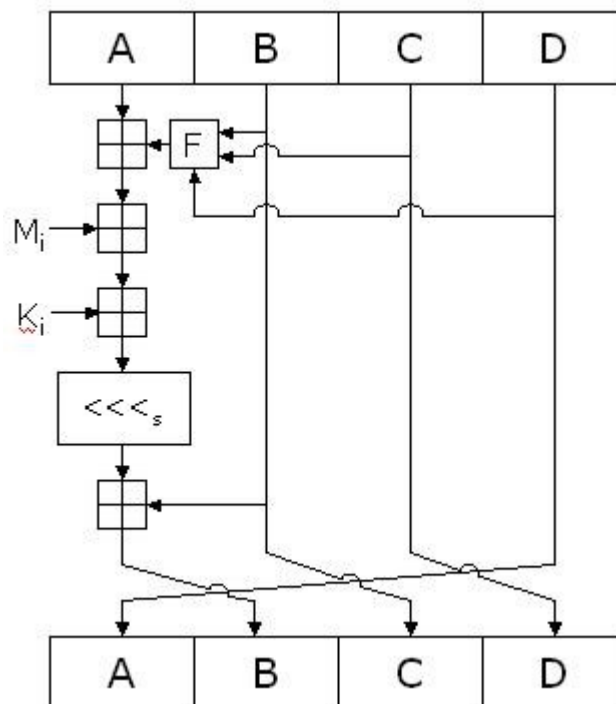

Fig 3.11 MD5

- The figure shows how the auxiliary function F is applied to the four buffers (A, B, C and D), using message word M_i and constant K_i. The item "<<<s" denotes a binary left shift by *s* bits.
- **The output**
- After all rounds have been performed, the buffers A, B, C and D contain the MD5 digest of the original input.

Digital Signature Techniques

- The national institute of standard and technology(NIST)published the (DSS Digital Signature Standard) standard as the federal information processing standard(FIPS) PUB 186 in 1991. Which was revised in 1993 and 1996. DSS makes use of the SHA-1

algorithm for calculating the message digest over an original message and uses the message digest to perform the digital signature .

- For this, DSS makes use of an algorithm, called as **Digital signature algorithm(DSA**). DSS is the standard and DSA is the actual algorithm.
- RSA is used for encryption of a message and for Digital Signature but DSA is used only for Digital Signature.

RSA and digital signatures:

- We have mentioned that RSA can be used for performing digital signatures. Let us understand how this works in a step-by-step fashion. For this M let us assume that the sender(A) wants to send a message M to the receiver(B)along with the digital signature(S)calculated over the message(M). **step 1:-** the sender(A)uses the SHA-1 message digest algorithm to calculate the message digest(MDI)over the original message(M).
- **Step 2:-** the sender(A) now encrypts the message digest with her private key. The output of this process is called as the digital signature(DS) of A.
- **Step 3:-** now the sender(A)sends the original message(M)along with the digital signature(DS) to the receiver(B).
- **Step 4**:- after the receiver (B) receives the original message (M) and the sender"s (A"s) digital signature, B uses the same message digest algorithm as was used by the A and calculates its own message digest(MD2).
- **Step 5:-** the receiver(B)now uses the sender"s (A"s)public key to decrypt (sometimes)also called as de-sign) the digital signature. Note that A had used her private key to encrypt her message digest(MDI) to form the digital signature. Therefore only A"s public key can be used to decrypt it. The output of this process is the original message digest as was calculated by A (MDI) in step 1.
- Step 6:- B now compares the following 2 message digests:
  MD2, which it had calculated in step 4
  MD1, which it retrieved from A"s digital signature in step 5.

  - If MD1=MD2, the following facts are established.
  - B accepts the original message (M) as the correct, unaltered, message from A.
  - Bis also assured that the message came from A and not from someone posing as A.

DIGITAL CERTIFICATE & PUBLIC KEY INFRASTRUCTURE

4.1 DIGITAL CERTIFICATES

- A digital certificate is an electronic "passport" that allows a person, computer or organization to exchange information securely over the Internet using the public key infrastructure (PKI). A digital certificate may also be referred to as a public key certificate.

- Just like a passport, a digital certificate provides identifying information is forgery resistant and can be verified because it was issued by an official, trusted agency. The certificate contains the name of the certificate holder, a serial number, expiration dates, a copy of the certificate holder's public key (used for encrypting messages and digital signatures) and the digital signature of the certificate-issuing authority (CA) so that a recipient can verify that the certificate is real.

To provide evidence that a certificate is genuine and valid, it is digitally signed by a root certificate belonging to a trusted certificate authority. Operating systems and browsers maintain lists of trusted CA root certificates so they can easily verify certificates that the CAs have issued and signed. When PKI is deployed internally, digital certificates can be self-signed.

Digital certificates, similar to identification cards, are electronic credentials that are used to certify the online identities of individuals, organizations, and computers. Certificates are issued and certified by CAs. PKIX-compliant public key infrastructures support industry standard X.509 version 3 certificates.
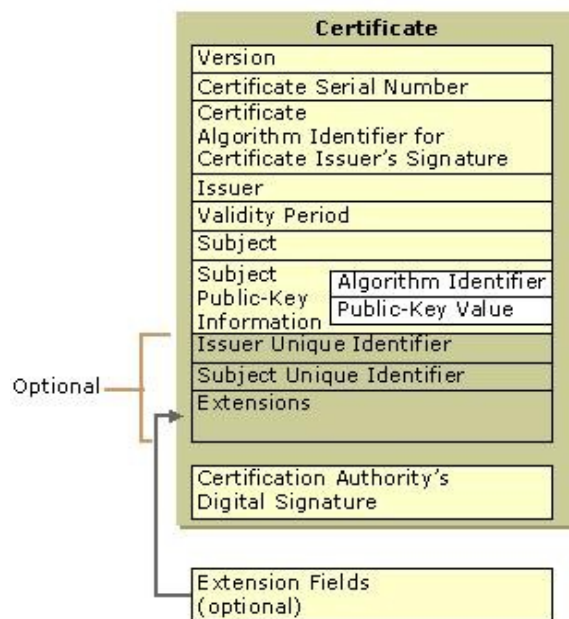

Fig 4.1 Digital Certificate

Contents of a typical digital certificate

- **Serial Number**: Used to uniquely identify the certificate.
- **Subject**: The person, or entity identified.
- **Signature Algorithm**: The algorithm used to create the signature.
- **Signature**: The actual signature to verify that it came from the issuer.
- **Issuer**: The entity that verified the information and issued the certificate.
- **Valid-From**: The date the certificate is first valid from.
- **Valid-To**: The expiration date.
- **Key-Usage**: Purpose of the public key (e.g. encipherment, signature, certificate signing...).
- **Public Key**: The public key.
- **Thumbprint Algorithm**: The algorithm used to hash the public key certificate.
- **Thumbprint** (also known as fingerprint): The hash itself, used as an abbreviated form of the public key certificate.

Certificate Authority

- Certificates are signed by the Certificate Authority (CA) that issues them. In essence, a CA is a commonly trusted third party that is relied upon to verify the matching of public keys to identity, e-mail name, or other such information.
- A certificate shows that a public key stored in the certificate belongs to the subject of that certificate. A CA is responsible for verifying the identity of a requesting entity before issuing a certificate. The CA then signs the certificate using its private key, which is used to verify the certificate. A CA's public keys are distributed in software packages such as Web browsers and operating systems, or they can also be added manually by the user.

Technical details of Digital Certificates.

- A standard called as X.509 defines the structure of a digital certificate. The International Telecommunication Union (ITU) came up with this standard in 1988. At that time, it was a part of another standard called as X.500. Since then X.509 was revised twice (in 1993 and again the 1995). The current version of the standard is Version 3, called as X.509V3. The Internet Engineering Task Force (IETF) pubished the RFC2459 for the X.509 standard in 1999.
- The contents of X.509 version 3 certificates are described in Table 4.1.
- **Table Description of X.509 Version 3 Certificate Contents**

Certificate Field	Description
Version	Version of the certificate format; for example, version 3.
Certificate Serial Number	The unique serial number that is assigned by the issuing CA. The CA maintains an audit history for each certificate so that certificates can be traced by their serial numbers. Revoked certificates also can be traced by their serial numbers.
Certificate Algorithm Identifier	The public key cryptography and message digest algorithms that are used by the issuing CA to digitally sign the certificate.
Issuer	The name of the issuing CA. The name can be listed in one or more of the following formats: X.500 directory name, Internet e-mail address, fully qualified domain name (FQDN), X.400 e-mail address, and URL.
Validity Period	The certificate's start and expiration dates. These define the interval

	during which the certificate is valid, although the certificate can be revoked before the designated expiration date.
Subject	The name of the subject (owner) of the certificate. The name can be listed in one or more of the following formats: X.500 directory name, Internet e-mail address, fully qualified domain name (FQDN), X.400 e-mail address, and URL.
Subject Public-Key Information	The public key and a list of the public key cryptography algorithms. The algorithms are for the tasks for which the public key set can be used, such as digital signing, secret key encryption, and authentication.
Issuer Unique Identifier	Optional information for uniquely identifying the issuer, when necessary.
Subject Unique Identifier	Optional information for uniquely identifying the subject, when necessary.
Extensions	Additional information that can be specified for optional use by public key infrastructures. Common extensions include a list of specific uses for certificates (for example, S/MIME secure mail or IPSec authentication), CA trust relationship and hierarchy information, a list of publication points for revocation lists, and a list of additional attributes for the issuer and subject.
Certification Authority's Digital Signature	The CA's digital signature, which is created as the last step in generating the certificate.

Table 4.1 the contents of X.509 version 3

Digital Certificate Creation

- There are two of the three parties involved in the digital certificate creation process, namely the subject (end user) and the issuer (CA). A third party is also (optionally) involved in the certificate creation and management.
- Since a CA can be overloaded with a variety of tasks such as issuing new certificates , maintaining the old ones ,revoking the ones that have become invalid for whatever reason, etc. The CA can delegate some of its tasks to this third party, called as a registration Authority (RA).The RA is an intermediate entity between the end users and the CA , which assists the CA in its day-to-day activities.
- The RA commonly provides following services.
- Accepting and verifying registration information about new users
- Generating keys on behalf on the end users
- Accepting and authorizing requests for key backups and recovery •
Accepting and authorizing the requests for certificate revocation

Certificate creation steps

- The creation of a digital certificate consists of several steps

- *Step1: Key generation* The action begins with the subject (i.e the user/organization) who wants to obtain a certificate. There are two different approaches for this purpose:
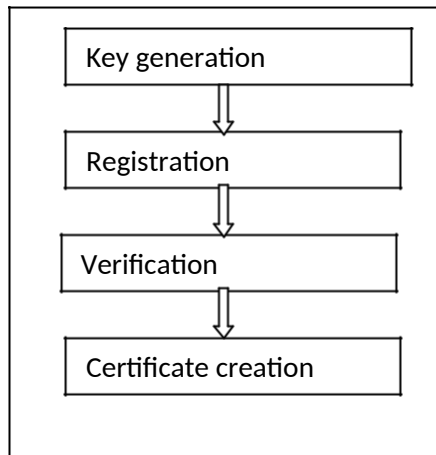
Fig 4.2 Digital certificate creation steps

(a) The subject can create a private key and public key pair using some software. This software is usually a part of the web browser or web server.

(b) Alternatively, the RA can generate a key pair on the subjects (users) behalf. This can happen in cases where either the user is not aware of the technicalities involved in the generation of a key pair or if a particular requirement demands that all the keys must be centrally generated and distributed by the RA for the ease of enforcing security policies and key management.

- **Step 2: Registration** This step is required only the user generates the key pair in the first step. If the RA generates the key pair on the users behalf, this step will also be a part of the first step itself.

- Assuming that the user has generated the key pair, the user now sends the public key and the all the evidence about herself to the RA. For this, the software provides a wizard in which the user enters data and when all data is correct, submits it.

- This data then travel s over the network/internet to the RA. The format for the certificate requests has been standardized and is called as **certificate signing request (CSR)** .This is one of the **public key cryptography standard** (PKCS), as we shall study later. CSR is also called as PKCS #10.

- The user must not send the private key to the RA- the user must retain it securely. In fact, as far as possible, the private key must not leave the user"s computer at all.

- **Step 3 verification** After the registration process is complete, the RA has to verify the users credentials. This verification is in to respects, as follow.

(a) Firstly, the RA needs to verify the user"s credentials such as the evidences provided are correct and that they are acceptable. If the user were actually an organization, then the RA would perhaps like to check the business records, historical documents and credibility proofs.

(b) The second check is to insure that the user who is requesting for the certificate does indeed possess the private key corresponding to the public key that is sent as a part of the certificate request to the RA. This is very important, because, there must be a record that the user possesses the private key corresponding to the given public key. This check is called as checking the **proof of possession (POP)**of the private key .there are many approaches to this

1. The RA can demand that the user must digitally sign her certificate signing Request (CSR) using her private key. If the RA can verify the signature (i. e . de-sign the CSR)

correctly using the public key of the user , the RA can believe that user indeed possesses the private key .

2. Alternatively, at this stage, the RA can create a random number challenge; encrypt it with the user"s public key and send the Encrypted challenge to the user. If the user can successfully decrypt the challenge using her private key, the RA can assume that the user possesses the right private key.

3. Thirdly, the RA can actually generate a dummy certificate for the user, encrypt it using the users public key and send it to the user. The user can decrypt it only if she can decrypt the encrypted certificate and obtain the plain text certificate.

- ***Step 4 certificate creation Assuming*** that all the steps so far have been successful , the RA passes on all the details of the user to the CA. The CA does its own verification (if required) and creates a digital certificate for the user. There are programs for creating certificates in the X.509 standard format. The CA sends the certificate to the user and also retains a copy of the certificate for its own record. The CA"s copy of the certificate is maintained in a certificate directory. This is a central storage location maintained by the CA. The contents of the certificate directory are similar to that of a telephone directory. This facilitates for a single-point access for certificate management and distribution.

- The CA then sends the certificate to the user. This can be attached to an e-mail or the CA can send an e-mail to the user, informing that the certificate is ready and that the user should download it from the CA"s site. The latter possibility is depicted.

- The user gets a screen, which informs the user that his digital certificate is ready and he should download it from the CA"s site.

- The digital certificate is actually in an unreadable format to human eyes . We can make little sense out of this certificate. However, an application program actually parses or interprets the certificate and shows us its details in a human- readable format.
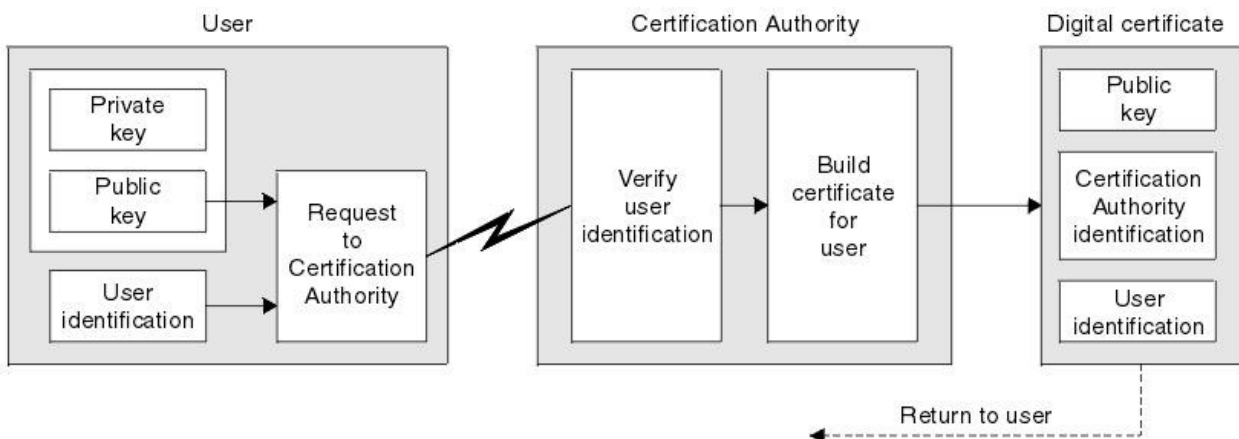

Fig 4.3 Creation of Digital certificate

Why Should We Trust Digital Certificate?

- The same reason you trust what is stated in a driver's license: endorsement by the relevant authority (Department of Transport) in the form of a difficult to forge signature or stamp of approval. Digital certificates are endorsed in a similar manner by a trusted authority empowered by law to issue them, appropriately known as the Certifying Authority or CA. The CA is responsible for vetting all applications for digital certificates, and once satisfied, "stamps" its difficult to forge digital signature on all the digital certificates it issues, attesting to their validity.

How can we verify a Digital Certificate?

- Having understood how the CA signs a digital certificate, let us now think how the verification of a certificate take place. Suppose we receive a digital certificate of a user, which we want to verify. What should we do for this ? Clearly, we need to verify the digital signature of the CA. Let us understand what steps are involved in this process, as

- The verification of a digital certificate consists of the following steps .
1. The user passes all fields except the last one of the received digital certificate to a message digest algorithm. This algorithm should be the same as the one used by the CA while signing the certificate, so the user here knows which algorithm is to be used.
2. The message digest algorithm calculates a message digest (hash) of all fields of the certificate, except for the last one. Let us call this message digest as MDI.
3. The user now extracts the digital signature of the CA from the certificate (remember, it is the last field in a certificate).
4. 4. The user de-signs the CA"s signature (i.e. the user decrypts the signature with the CA"s public key).
5. This produces another message digest , as would have been calculated by the CA during the signing of the certificate (i.e. before it encrypted the message digest with its private key to create its digital signature over the certificate).
6. Now the user compares the message digest it calculated (MDI) with the one, which is the result of de-signing the CA"s signature (MD2). If the two match, i.e. if MD1 = MD2, the user is convinced that the digital certificate was indeed signed by the CA with its private key. If this comparison fails, the user will not trust the certificate and reject it.

Certificate Hierarchies and self Signed Digital Certificate

- CAs are hierarchical in structure. There are generally three types of hierarchies, and they are denoted by the number of tiers.

  Single/One Tier Hierarchy

Fig 4.4 Single/One tier hierarchy

- A single tier Hierarchy consists of one CA. The single CA is both a Root CA and an Issuing CA. A Root CA is the term for the trust anchor of the PKI. Any applications, users, or computers that trust the Root CA trust any certificates issued by the CA hierarchy.
- The Issuing CA is a CA that issues certificates to end entities. For security reasons, these two roles are normally separated. When using a single tier hierarchy they are combined.
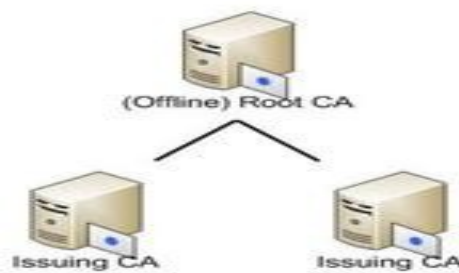
Two Tier Hierarchy

Fig 4.5 Two tier hierarchy

- A two tier hierarchy is most common. In some ways it is a compromise between the One and Three Tier hierarchies. In this design there is a Root CA that is offline, and a subordinate issuing CA that is online. The level of security is increased because the Root CA and Issuing CA roles are separated. But more importantly the Root CA is offline, and so the private key of the Root CA is better protected from compromise.
- It also increases scalability and flexibility. This is due to the fact that there can be multiple Issuing CA"s that are subordinate to the Root CA. This allows you to have CA"s in different geographical location, as well as with different security levels.
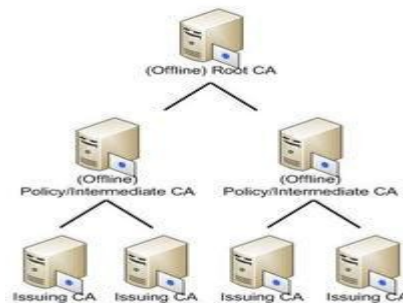
Three Tier Hierarchy

Fig 4.6 Three tier hierarchy

- Specifically the difference between a Two Tier Hierarchy is that second tier is placed between the Root CA and the issuing CA. The placement of this CA can be for a couple different reasons. The first reason would be to use the second tier CA as a Policy CA.
- In other words the Policy CA is configured to issue certificates to the Issuing CA that is restricted in what type of certificates it issues. The Policy CA can also just be used as an administrative boundary.
- In other words, you only issue certain certificates from subordinates of the Policy CA, and perform a certain level of verification before issuing certificates, but the policy is only enforced from an administrative not technical perspective.The other reason to have the second tier added is so that if you need to revoke a number of CAs due to a key compromise, you can perform it at the Second Tier level, leaving other "branches from the root" available. It should be noted that Second Tier CAs in this hierarchy can, like the Root, be kept offline.

Self-signed Certificate

- In cryptography and computer security, a **self-signed certificate** is an identity certificate that is signed by the same entity whose identity it certifies.

- This term has nothing to do with the identity of the person or organization that actually performed the signing procedure. In technical terms a self-signed certificate is one signed with its own private key.
- In typical public key infrastructure (PKI) arrangements, a digital signature from a certificate authority (CA) attests that a particular public key certificate is valid (i.e., contains correct information). Users, or their software on their behalf, check that the private key used to sign some certificate matches the public key in the CA's certificate. Since CA certificates are often signed by other, "higher-ranking," CAs, there must necessarily be a highest CA, which provides the ultimate in attestation authority in that particular PKI scheme.
- Obviously, the highest-ranking CA's certificate can't be attested by some other higher CA (there being none), and so that certificate can only be "self-signed." Such certificates are also termed **root certificates**.
- Clearly, the lack of mistakes or corruption in the issuance of such certificates is critical to the operation of its associated PKI; they should be, and generally are, issued with great care.

Cross Certification

- Cross certification enables entities in one public key infrastructure (PKI) to trust entities in another PKI. This mutual trust relationship is typically supported by a cross-certification agreement between the certification authorities (CAs) in each PKI. The agreement establishes the responsibilities and liability of each party.
- A mutual trust relationship between two CAs requires that each CA issue a certificate to the other to establish the relationship in both directions. The path of trust is not hierarchal (neither of the governing CAs is subordinate to the other) although the separate PKIs may be certificate hierarchies.
- After two CAs have established and specified the terms of trust and issued certificates to each other, entities within the separate PKIs can interact subject to the policies specified in the certificates.

Revoke A Digital Certificate

- When a certificate authority (CA) generates a certificate, that certificate is valid for a specific amount of time. The expiration date is part of the certificate itself. Similar to a driver's license that can be suspended even if it has not expired, a certificate can be revoked before it has expired. For example, a certificate can be revoked if an employee leaves a company or moves to a new position in the same company.
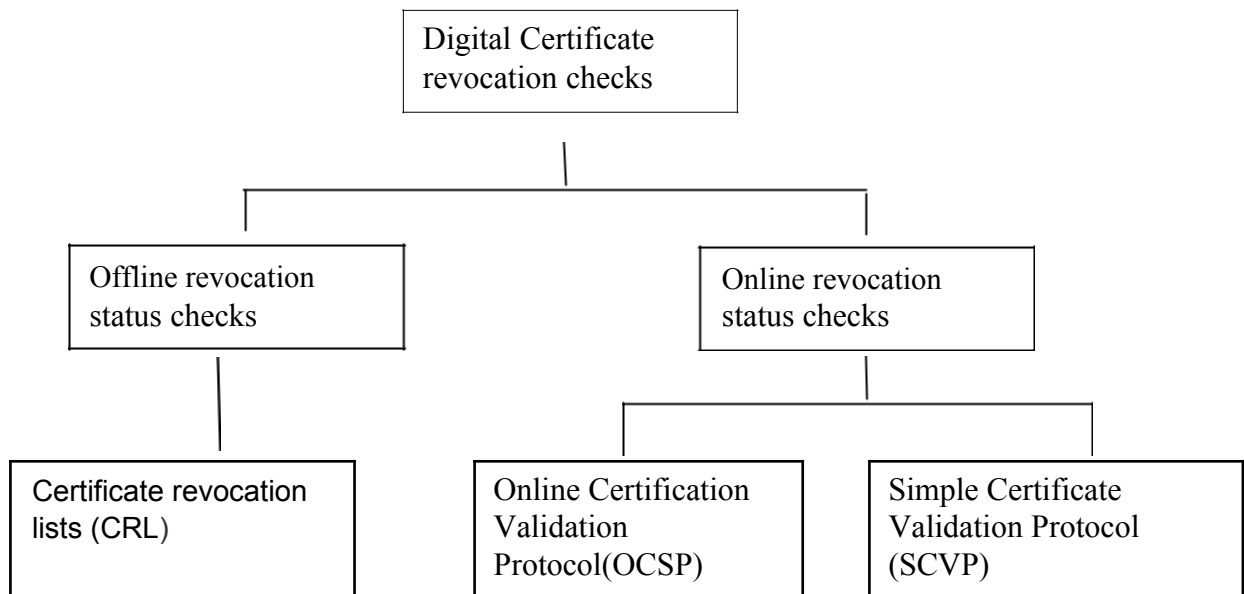
```
                    ┌─────────────────────┐
                    │ Digital Certificate │
                    │ revocation checks   │
                    └─────────────────────┘
                              │
              ┌───────────────┴───────────────┐
  ┌─────────────────────┐         ┌─────────────────────┐
  │ Offline revocation  │         │ Online revocation   │
  │ status checks       │         │ status checks       │
  └─────────────────────┘         └─────────────────────┘
            │                        ┌───────┴───────┐
  ┌─────────────────────┐  ┌──────────────────┐  ┌──────────────────┐
  │ Certificate         │  │ Online           │  │ Simple           │
  │ revocation          │  │ Certification    │  │ Certificate      │
  │ lists (CRL)         │  │ Validation       │  │ Validation       │
  │                     │  │ Protocol(OCSP)   │  │ Protocol (SCVP)  │
  └─────────────────────┘  └──────────────────┘  └──────────────────┘
```

Fig 4.7 Revoke a digital certificate

- There are two different states of revocation

- Revoked: A certificate is irreversibly revoked if, for example, it is discovered that the certificate authority (CA) had improperly issued a certificate, or if a private-key is thought to have been compromised. Certificates may also be revoked for failure of the identified entity to adhere to policy requirements, such as publication of false documents, mis-representation of software behavior, or violation of any other policy specified by the CA operator or its customer. The most common reason for revocation is the user no longer being in sole possession of the private key (*e.g.*, the token containing the private key has been lost or stolen).

- Hold: This reversible status can be used to note the temporary invalidity of the certificate (*e.g.*, if the user is unsure if the private key has been lost). If, in this example, the private key was found and nobody had access to it, the status could be reinstated, and the certificate is valid again, thus removing the certificate from future CRLs.

Offline Certificate Revocation Status Checks

- If, for example, you are using the certificate in sensitive business applications you may want a way to revoke that certificate. There are several ways to do this. One approach is to have the CA revoke any certificate that it created. After one or more certificates have been revoked, the CA generates a certificate revocation list (CRL) that can be checked during the authentication process.The CRL, which is signed by the CA to prevent the CRL from being tampered with, contains the following:
- A list of the serial numbers for the certificates that have been revoked
- The name of the CA who is the issuer of the certificates and the CRL.

- Once CRLs are created, they must be sent to the relevant nodes that validate certificates generated by this CA. In the Internet environment, CRLs are not practical because there is no way to send them to the entire world. However, in the Intranet environment, you have a limited number of nodes at your disposal so you can populate CRLs, if needed, using File Transfer Protocol

- A CRL is generated and published periodically, often at a defined interval. A CRL can also be published immediately after a certificate has been revoked. The CRL is always issued by the CA which issues the corresponding certificates. All CRLs have a lifetime during which they are valid; this timeframe is often 24 hours or less. During a CRL's validity period, it may be consulted by a PKI-enabled application to verify a certificate prior to use.

Certificate Revocation List (CRL) –

A CRL is really just a list of revoked certificate serial numbers that has been digitally signed and time-stamped and placed in a public in a repository such as a web site. Applications that use certificates have the option of checking the CDP repositories for CRLs. Many applications ignore CRLs altogether while other applications check CRLs and choose what to do if the certificate has been revoked as is the case with many web browsers.

- There are different kinds of CRLs that are generally published on repositories known as CRL distribution points or CDPs:

- **Full CRL** or **Base CRL** - The most common and widely supported are the full CRLs also known as base CRLs which contain serial numbers of all revoked certificates for a particular Certificate Authority (CA).
- **Delta CRL** – Delta CRLs contain the list of serial numbers of only certificates that have been revoked since the last Base CRL was published.

One of the challenges with designing a PKI is to determine the best publication interval, and there are several factors to consider.

CRL Publishing considerations:

- Publishing a base CRL more frequently, revoked certificates could be more quickly known, but the CRLs are downloaded more often, and consequently as the CRLs grow can generate more traffic for the full CRL downloading by all clients.
- Publishing CRLs less often can increase the latency before a client becomes aware of a newly revoked certificate, but may reduce overall network traffic because CRLs are downloaded less often.

Delta CRLs which are much smaller than base CRLs generally are defined in RFC 2380 and allow base CRLs to be downloaded at intervals further apart with more frequent downloads of the delta CRL. This allows for more frequent updates to the known revoked certificates without the necessity to download the full CRL very often. Not all devices or applications recognize delta CRLs.

Online Certificate Status Protocol (OCSP)

- OCSP is an Internet protocol used for obtaining the revocation status of an X.509 digital certificate. It is described in RFC 6960 and is on the Internet standards track. It was created as an alternative to certificate revocation lists (CRL), specifically addressing certain problems associated with using CRLs in a public key infrastructure (PKI).

Messages communicated via OCSP are encoded in ASN.1 and are usually communicated over HTTP. The "request/response" nature of these messages leads to OCSP servers being termed *OCSP responders*.

- **OCSP** - for real-time validation, the **Online Certificate Services Protocol** (OCSP) is an HTTP protocol that acts as an intermediary to responder to clients that support the protocol.

- The OCSP response is a digitally signed response for the certificate status, but the response size does not change regardless of the number of revoked certificates. On the back end, the OCSP responder generally relies on CRLs.

- The advantage with OCSP over CRLs, is that in the event of a revocation that requires near immediate response a new CRL can be published and the OCSP responder can be configured to get the new CRL at a pre-determined interval (i.e. a few minutes) rather than waiting on the next update cached in the CRL.

1. The CA provides a server, called as an OCSP responder. This server contains the latest certificate revocation information. The requestor (client) has to send a query (called as OCSP request) about a particular certificate to check whether it is revoked or not. The underlying protocol for OCSP is most commonly HTTP, although other application-layer protocols (such as SMTP) can be used. Actually, this is technically not completely right. In practice, the OCSP request contains the OCSP protocol version, the service requested and one or more certificate identifiers (which, in turn, consist of a message digest of the issuer"s name, a message digest of the issuer"s public key and the certificate serial number). However, we shall ignore this for the sake of simplicity.

2. The OCSP responder consults the server"s X.500 directory (in which the CA continuously feeds the certificate revocation information) to see if the particular certificate is valid or not.

- Based on the result of the status check from the X.500 directory lookup, the OCSP responder sends back a digitally signed OCSP response for each of the certificates in the original request to the client. This response can take one of the three forms, namely, Good, Revoked

Advantages over CRL

- OCSP may be used to satisfy some of the operational requirements of providing more timely revocation information than is possible with CRLs.
- OCSP removes the need for clients to retrieve the (sometimes very large) CRLs themselves, leading to less network traffic and better bandwidth management.
- Clients do not need to parse CRLs themselves, saving client-side complexity.
- An OCSP responder may implement billing mechanisms to pass the cost of validation transactions to the seller, rather than buyer.
- CRLs may be seen as analogous to a credit card company's "bad customer list" which is a bad public exposure.

Simple Certificate Validation Protocol(SCVP)

- SCVP is Iin draft stage as of the current writing.SCVP is an online certificate status reporting protocol, designed to deal with drawback of OCSP. Since SCVP is similar to OSCP

Certificate Types

- Not all digital certificates have the same status and cost. Depending on the requirement, these differ for instance a digital certificate can be used by a user only for encrypting message, but not for digitally signing any messages.
- Generally, the certificate types can be classified as follows:
- **Email certificates:** Email certificates include the user"s email id. This is used to verify that the signer of an email message has an email id that is the same as it appears in that user"s certificate.
- **Server-side SSL certificates:** These certificates are useful for merchants who want to allow buyers to purchase goods or services from their online Website. We shall discuss this in detail later. Since a misuse of this certificate can cause serious damages, such certificates are issued only after a careful scrutiny of the merchant"s credentials.
- **Client-side SSL certificates:** These certificates allow a merchant (or any other server-side entity) to verify a client (browser-side entity). We shall discuss these certificates in detail later.
- **Code-signing certificates:** many people do not like to download client-side code such as Java applets or ActiveX controls, because of the inherent risks associated with them. In order to alleviate these concerns, the code (i.e. the Java applets or ActiveX controls) can be signed by the signer. When a user hits a Web page contains such pieces of code, signed by the appropriate developer/organization and whether the user would like to trust that developer/organization. If the user responds affirmatively, the Java applets or ActiveX controls are downloaded and get executed on the browser. However, if the user rejects the offer, the processing ends there. It must be noted that mere signing of code does not make it safe –the code cause havoc. It simply specifies where the code originates.

Roaming certificates

- Digital certificate so far, it should be quite obviously that certificate greatly help in establishing trust about the users, based on their public keys. people to move around and yet be able to perform digital transaction, such as encryption and signing. This means that the digital certificate must also be mobile for such users. Smartcard is obviously one technology for making this possible
- A better solution of roaming certificate is now in use .A third party provides this solution to an organization"s user s or individual users. This works is as follows.
1- The user"s digital certificate and private keys along with the user ids and passwords are stored in the database of a central secure server, called as the credential server, as shown in figure.
2- When a user moves around and then longs into her computer, she authenticates herself using the id and password to the credential server over the internet.
3- The credential server verifies the user id and password, using its credential database. If the user is successfully authenticated, the credential server sends the user her digital certificate and private key file, as shown in figure.

4.2 Private key management

Protecting private key

- In many situations, the private key of the user might be required to be transported from one location to another. For instance, suppose that the user wants to change her PC. To handle these situations, there is a cryptography standard by the name PKCS#12.This

allows a user to export her digital certificate and private key in the form of a computer file. Obviously, the certificate and the private key must be protected as they are moved to another location. for this ,the PKCS#12 standard ensures that they are encrypted using a symmetric key which is derived from the user's private key protection password.

Multiple key pairs

- The PKI approach also recommends that in serious business applications, user should posses multiple digital certificates, which also means multiple key pairs. The need for this is that one certificate could be strictly used for signing anther for encryption. This ensures that the loss of one of the private keys does not affect the complete operations of the user. The following guidelines are generally helpful:
- The private key that is used for digital signing(non-repudiation)must be backed up or archived after it expires. it must be destroyed. This ensures that it is not used by someone else for signing on behalf of the person at a future date(although chances are that this will be detected by CRL/OCSP checks or certificate expiry date checks, you cannot say this with a 100% guarantee).
- In contrast the private key used for encryption/decryption must be backed up after its expiry, so that encrypted information can be recovered even at a later date.

☁ **Mechanism for protecting private keys**

mechanism	description
password protection	This is the simplest and most common mechanism to prote key. The private key is stored on the hard disk of the uses computer as a disk file. This file can be accessed only with the help of a password or a personal identification number(PIN).since anyone who can guess the password correctly can access the private key, this is considered as the least secure Method of protecting a private key. PCMCIA cards The personal computer memory card international association(PCMCIA)cards are actually chip cards. The private key is stored on such a card ,which means that it need not be on the user"s hard disk ,this Reduces the chances of it being stolen. however, for a cryptographic application such as signing or encryption, the key must travel from the PCMCIA card to the Memory of the user"s computer. Therefore, there is still scope for it being Captured from there by an attacker.
PCMCIA cards	The personal computer memory card international association (PCMCIA)cards are actually chip cards. The private key is stored on such a card ,which means that it need not be on the user"s hard disk. This reduces the chances of it being stolen. however, for a cryptographic application such as signing or encryption, the key must travel from the PCMCIA card to the Memory of the user"s computer Therefore, there is still scope for it being Captured from there by an attacker.

tokens	A token stores the private key in an encrypted format. To decrypt and access it the user must provide a one-time password(which means that the password is valid only for that particular access, next time, this password becomes invalid and another must be used) we shall later discuss how this works. this is a more secure method.
Biometrics	The private key is associated with a unique characteristics of an individual(such as fingerprint, retina scan or voice comparison)This is similar in concept to the tokens, but here the user need not carry anything with him, unlike the token.
smart cards	In a smart card, the private key of the user is stored in a tamperproof card. this card also contains a computer chip, which can perform cryptographic functions such as signing and encryption .The biggest benefit of this scheme is that the private key never leaves the smart card. Thus,the scope for its compromise is tremendously reduced. The disadvantage of this scheme is that the user needs to carry the smart card with her and available to access it.

Table 4.2 mechanisms for protecting private key

KEY UPDATE

Good security practices demand that the key pairs should be updated periodically. this is because over time, keys become susceptible to analysis attacks. Causing a digital certificate to expire after a certain date ensures this. This requires an update to the key pair. The expiry of a certificate can be dealt with in one of the two following ways:

- The CA reissues a new certificate based on the original key pair (of course, this is not recommended unless there is an all-around confidence in the strength of the original key pair).
- .A fresh key pair is generated and the CA issues a new certificate based on that the new pair. The key update process itself can be handled in two ways, as follows:
- In the first approach, the end user has to detect that certificate is about to expire and request the CA to issue a new one.
- In the other approach,the expiry date of the certificate is automatically checked every time it is used and as soon as it is about to expire,its renewal request is sent to the CA.forthis,special systems need to be in place.

KEY ARCHIVAL

- The CA must plan for and maintain the history of the certificate and the keys of its users. for instance, suppose that someone approaches the CA of Alice, requesting the CA to make Alice"s digital certificate available, as was used three years back to sign a legal document for verification purposes. if the CA has not archived the certificates, how can the CA provide this information? This can cause serious legal problems. Therefore, key archival is a very significant aspect of any PKI solution.

4.3 PKIX Model

- Management protocols are the protocols that are required to support on–line interactions between PKI user and management entities. The possible set of functions that can be supported by management protocols is
- registration of entity, that takes place prior to issuing the certificate .
- initialisation, for example generation of key–pair.
- certification, the issuance of the certificate .
- key–pair recovery, the ability to recover lost keys .
- key–pair update, when the certificate expires and a new key–pair and certificate have to be generated.
- revocation request, when an authorised person advices the CA to include a specific certificate into the revocation list
- cross-certification, when two CAs exchange information in order to generate a cross–certificate .
  - The Certificate Policies and the Certificate Practice Statements are recommendations of documents that will describe the obligations and other rules with regard the usage of the Certificate.

PKI functionality

Functionality
Registration
Initialisation
Certification
Key–pair recovery
Key generation
Key update
Key expiry
Key compromise
Cross certification
Revocation
Certificate and Revocation Notice Distribution and Publication

Table 4.3 PKI functionality

PKIX Architectural Model

- PKIX is working on the following five areas.

1. Profiles of X.509 v3 Public Key Certificates and X.509 v2 Certificate Revocation Lists (CRLs).

It describes the basic certificate fields and the extensions to be supported for the Certificates and the Certificate Revocation Lists. Then, it talks about the basic and extended Certificate Path Validation. Finally, it covers the supported cryptographic algorithms.

2. Management protocols.

   First, it discusses the assumptions and restrictions of the protocols. Then, it provides the data structures used for the PKI management messages and defines the functions that conforming implementations must carry out. Finally, it describes a simple protocol for transporting PKI messages.

3. Operational protocols.

   Currently they describe how LDAPv2, FTP and HTTP can be used as operational protocols.

4. Certificate policies and Certificate Practice Statements.

   The purpose of this document is to establish a clear relationship between certificate policies and CPSs, and to present a framework to assist the writers of certificate policies or CPSs with their tasks. In particular, the framework identifies the elements that may need to be considered in formulating a certificate policy or a CPS. The purpose is not to define particular certificate policies or CPSs, per se.

5. Time–stamping and data–certification/validation services.

   There are no RFCs on these services yet, as the documents are still classified as Internet Drafts.

   The time–stamping services define a trusted third–party that creates time stamp tokens in order to indicate that a datum existed at a particular point in time. The data certification and validation services provide certification of possesion of data and claim of possesion of data, and validation of digitally signed documents and certificates.

   • The relevant Request For Comments (RFC) documents are depicted in the following table

4.4 Public key cryptography standards

   • In cryptography, **PKCS** is a group of **public-key cryptography** standards devised and published by RSA Security Inc, starting in the early 1990s. The company published the standards to promote the use of the cryptography techniques to which they had patents, such as the RSA algorithm, the Schnorr signature algorithm and several others. Though not industry standards (because the company retained control over them), some of the standards in recent years[when?] have begun to move into the "standards-track" processes of relevant standards organizations such as the IETF and the PKIX working-group.

PKCS Standards Summary			
	Version	Name	Comments
PKCS #1	2.1	RSA Cryptography Standard[1]	See RFC 3447. Defines the mathematical properties and format of RSA public and private keys (ASN.1-encoded in clear-text), and the basic algorithms and encoding/padding schemes for performing RSA encryption, decryption, and producing and verifying signatures.
PKCS #2	-	*Withdrawn*	No longer active as of 2010. Covered RSA encryption of message digests; subsequently merged into PKCS #1.
PKCS #3	1.4	Diffie–Hellman Key Agreement Standard[2]	A cryptographic protocol that allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communications channel.
PKCS #4	-	*Withdrawn*	No longer active as of 2010. Covered RSA key syntax; subsequently merged into PKCS #1.
PKCS #5	2.0	Password-based Encryption Standard[3]	See RFC 2898 and PBKDF2.
PKCS #6	1.5	Extended-Certificate Syntax Standard[4]	Defines extensions to the old v1 X.509 certificate specification. Obsoleted by v3 of the same.
PKCS #7	1.5	Cryptographic Message Syntax Standard[5]	See RFC 2315. Used to sign and/or encrypt messages under a PKI. Used also for certificate dissemination (for instance as a response to a PKCS#10 message). Formed the basis for S/MIME, which is as of 2010 based on RFC 5652, an updated Cryptographic Message Syntax Standard (CMS). Often used for single sign-on.
PKCS #8	1.2	Private-Key Information Syntax Standard[6]	See RFC 5208. Used to carry private certificate keypairs (encrypted or unencrypted).
PKCS #9	2.0	Selected Attribute Types[7]	See RFC 2985. Defines selected attribute types for use in PKCS #6 extended certificates, PKCS #7 digitally signed messages, PKCS #8 private-key information, and PKCS #10 certificate-signing requests.
PKCS #10		Certification Request Standard[8]	See RFC 2986. Format of messages sent to a certification authority to request certification of a public key. See certificate signing request.
PKCS #11	2.30	Cryptographic Token Interface[9]	Also known as "Cryptoki". An API defining a generic interface to cryptographic tokens (see also Hardware Security Module). Often used in single sign-on, public-key cryptography and disk encryption[10] systems. RSA Security has turned over further development of the PKCS#11 standard to the OASIS PKCS 11 Technical Committee.
PKCS #12		Personal Information Exchange Syntax	See RFC 7292. Defines a file format commonly used to store private keys with accompanying public key certificates, protected with a password-based symmetric key. PFX is a

PKCS			
	Standard[11]		predecessor to PKCS #12. This container format can contain multiple embedded objects, such as multiple certificates. Usually protected/encrypted with a password. Usable as a format for the Java key store and to establish client authentication certificates in Mozilla Firefox. Usable by Apache Tomcat.
PKCS #13	–	Elliptic Curve Cryptography Standard	*(Apparently abandoned, only reference is a proposal from 1998.)*[12]
PKCS #14	–	Pseudo-random Number Generation	*(Apparently abandoned, no documents exist.)*
PKCS #15	1.1	Cryptographic Token Information Format Standard[13]	Defines a standard allowing users of cryptographic tokens to identify themselves to applications, independent of the application's Cryptoki implementation (PKCS #11) or other API. RSA has relinquished IC-card-related parts of this standard to ISO/IEC 7816-15.[14]

Table 4.4 Public key cryptography standards

Internet security protocols

5.1 Basic concept

5.2 Secure socket layer

5.3 Transport layer security

5.4 Secure Hyper text transfer protocol(SHTTP)

5.5 Time stamping protocol (TSP)

5.6 Secure electronic transaction (SET)

5.1 BASIC CONCEPT

Static Web Page

- A web page created by a application developer and stored on a web server.
- Whenever any user request for that page, the web server sends back the same page without performing any addition processing.
- All it does is to locate that page on its hard disk and add HTTP headers, and Send back an HTTP response.
- Thus the contents of the Static web page do not change depending upon the request.
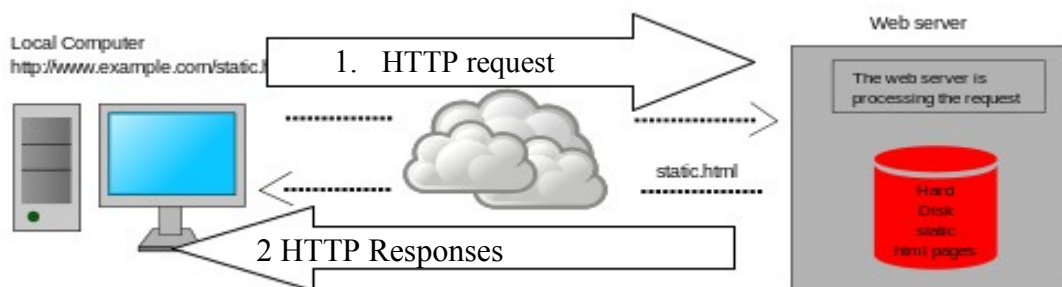
(Fig 5.1 Static Web Page)

Dynamic Web Page

- The contents of a dynamic web page can vary depending on a number of parameters.
- When a user request for a dynamic web page, the web server cannot simply send back an HTML page as in case of a static web page.
- Here the web server actually invokes a program that resides on its hard disk. The program might run database perform transaction processing etc. then send the HTTP response back to the web server.
- Dynamic web pages involve server side programming.
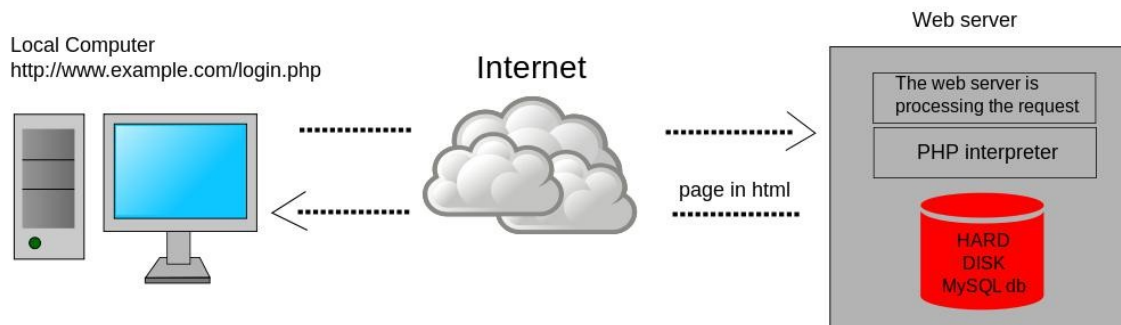
Fig 5.2 Dynamic Web Page

Active Web Page

- With the arrival of the programming language java, active web page became quite popular.
- The idea behind active web page is actually quite simple. When a client sends an HTTP request for an active web page, the web page server sends back an HTTP response that contains an HTML page as usual.
- In addition, the HTML page also contains a small program that executes on the client computer inside the web browser.
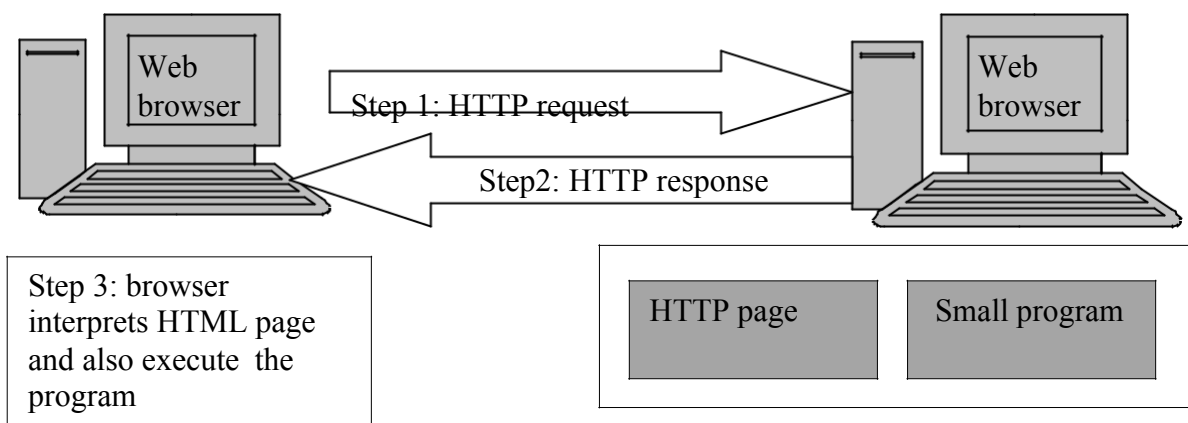
Fig 5.3 Active web page

TCP/IP protocol suite

- Communications between computers on a network is done through protocol suits. The most widely used and most widely available protocol suite is TCP/IP protocol suite. Each layer of the TCP/IP has a particular function to perform and each layer is completely separate from the layer(s) next to it.
- The communication process that takes place, at its simplest between two computers, is that the data moves from layer5 to 3 to 2 then to 1 and the information sent arrives at the second system and moves from 1 to 2 to 3 to 4 and then finally to layer 5. The 5 layers are as follows :-

1. Application layer
2. Transport layer
3. Network layer
4. Data link layer
5. Physical Layer

Application layer

- This is the top layer of TCP/IP protocol suite. This layer includes applications or processes that use transport layer protocols to deliver the data to destination computers.
- At each layer there are certain protocol options to carry out the task designated to that particular layer. So, application layer also has various protocols that applications use to communicate with the second layer, the transport layer. Some of the popular application layer protocols are :

- HTTP (Hypertext transfer protocol)
- FTP (File transfer protocol)
- SMTP (Simple mail transfer protocol)
- SNMP (Simple network management protocol) etc

Transport Layer

- This layer provides backbone to data flow between two hosts. This layer receives data from the application layer above it. There are many protocols that work at this layer but the two most commonly used protocols at transport layer are TCP and UDP.
- TCP is used where a reliable connection is required while UDP is used in case of unreliable connections.

Network Layer

- This layer is also known as Internet layer.
- The main purpose of this layer is to organize or handle the movement of data on network. By movement of data, we generally mean routing of data over the network.
- The main protocol used at this layer is IP. While ICMP(used by popular „ping" command) and IGMP are also used at this layer.

Data Link Layer

- This layer is also known as network interface layer.
- It provides error control and framing.
- Some of the famous protocols that are used at this layer include ARP(Address resolution protocol), PPP(Point to point protocol) etc.

Physical Layer

- This layer specifies the characteristics of the hardware to be used in the network.
- For example it specifies the characteristic of communication media.
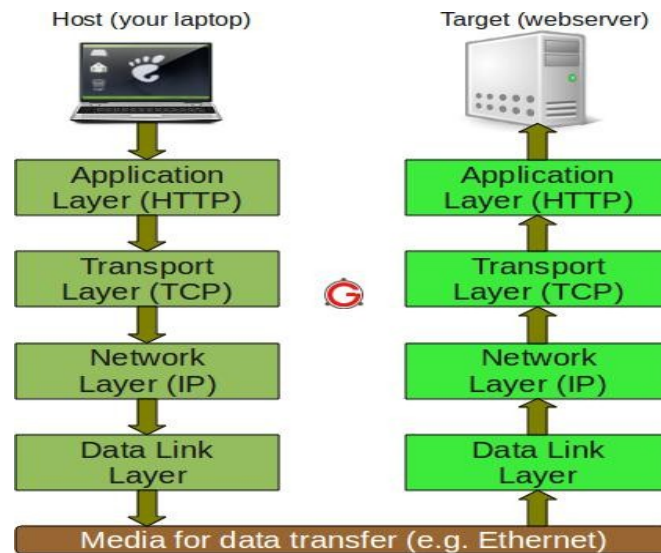
Fig 5.4 TCP/IP protocol suite

1.2 Secure socket layer

- The Secure Sockets Layer (SSL) is a computer networking protocol that manages server authentication, client authentication and encrypted communication between servers and clients.
- SSL uses a combination of public-key and symmetric-key encryption to secure a connection between two machines, typically a Web or mail server and a client machine, communicating over the Internet or an internal network.
- Using the OSI reference model as context, SSL runs above the TCP/IP protocol, which is responsible for the transport and routing of data over a network, and below higher-level protocols such as HTTP and IMAP, encrypting the data of network connections in the application layer of the Internet Protocol suite.
- The Transport Layer Security (TLS) protocol evolved from SSL and has largely superseded it, although the terms SSL or SSL/TLS are still commonly used; SSL is often used to refer to what is actually TLS.
- The combination of SSL/TLS is the most widely deployed security protocol used today and is found in applications such as Web browsers, email and basically any situation where data needs to be securely exchanged over a network, like file transfers, VPN connections, instant messaging and voice over IP.

The Position of SSL in TCP/IP Protocol Suite
- SSL can be conceptually considered as an additional layer in the TCP/IP protocol suite. The SSL layer located between the application layer and transport layer.
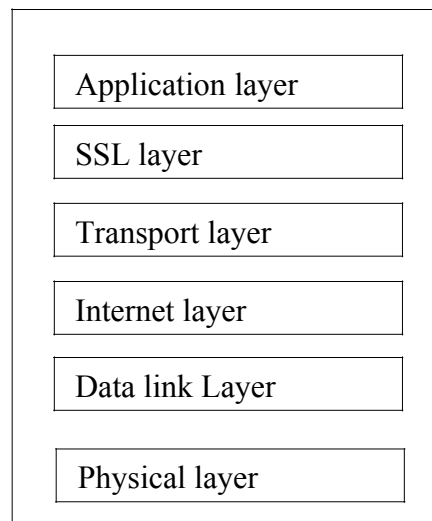
Fig 5.5 The Position of SSL in TCP/IP Protocol Suite

How ssl works

- As mentioned, the Secure Sockets Layer (SSL) is a method for providing security for web based applications.
- It is designed to make use of TCP to provide a reliable end to-end secure service.
- SSL has 3 sub protocol
    1. Handshake Protocol.
    2. Record Protocol.
    3. Alert Protocol.

Handshake Protocol

- This is the most complex part of SSL and allows the server and client to authenticate each other and to negotiate an encryption and MAC algorithm and cryptographic keys to be used to protect data sent in an SSL record.
- This protocol is used before any application data is sent. It consists of a series of messages exchanged by the client and server. Each message has three fields:

- Type (1 byte): Indicates one of 10 messages such as "hello request"
- Length (3 bytes): The length of the message in bytes.
- Content (_ 0 byte): The parameters associated with this message such version of SSL being used.
- The Handshake Protocol is consists of four phases:

1. **Establish security capabilities** including protocol version, session ID, cipher suite, compression method and initial random numbers. This phase consists of the client hello and server hello messages which contain the following (this is for the client however it"s a little different for the server):
 • Version: The highest SSL version understood by client

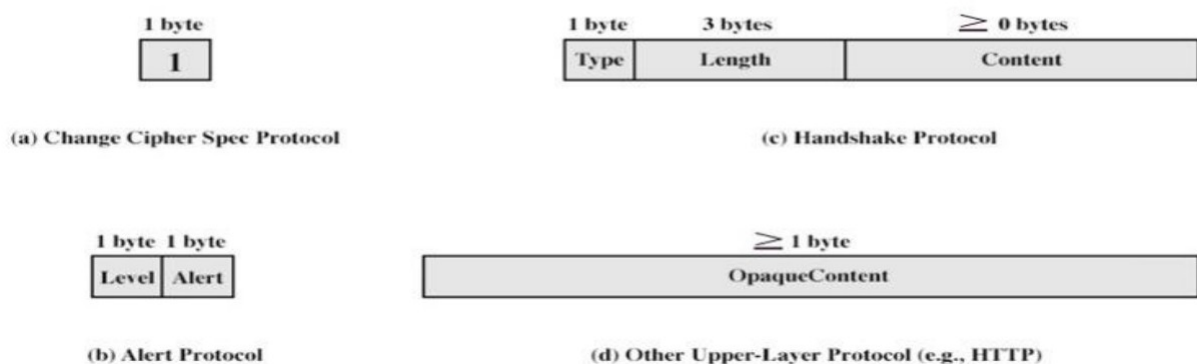| Message Type	Parameters
hello_request	null
client_hello	version, random, session id, cipher suite, compression method
server_hello	version, random, session id, cipher suite, compression method
certificate	chain of X.509v3 certificates
server_key_exchange	parameters, signature
certificate_request	type, authorities
server_done	null
certificate_verify	signature
client_key_exchange	parameters, signature
finished	hash value

1 byte

1

(a) Change Cipher Spec Protocol

1 byte   3 bytes   ≥ 0 bytes

Type	Length	Content

(c) Handshake Protocol

1 byte 1 byte

Level	Alert

(b) Alert Protocol

≥ 1 byte

OpaqueContent

(d) Other Upper-Layer Protocol (e.g., HTTP)

Fig 5.6 SSL Handshake Protocol

SSL record protocol payload.
• Random: 32-bit timestamp and 28 byte nonce.
• Session ID: A variable length session identifier.
• CipherSuite: List of cryptoalgorithms supported by client in decreasing order of preference. Both key exchange and CipherSpec (this includes fields such as CipherAlgorithm, MacAlgorithm, CipherType, HashSize, Key Material and IV Size) are defined.
• Compression Method: List of methods supported by client.

2. **Server may send certificate, key exchange**, and request certificate it also signals end of hello message phase. The certificate sent is one of a chain of X.509 certificates discussed earlier in the course. The server key exchange is sent only if required. A certificate may be requested from the client if needs be by certificate request.

3. **Client authentication and Key exchange** upon receipt of the server done message, the client should verify that the server provided a valid certificate, if required, and check that the server hello parameters are acceptable. If all is satisfactory, the client sends one or more messages back to the server. The client sends certificate if requested (if none available then it sends a no certificate alert instead). Next the client sends client key exchange message . Finally, the client may send certificate verification.

4. **Finish** Change cipher suite and finish handshake protocol. The secure connection is now setup and the client and server may begin to exchange application layer data.
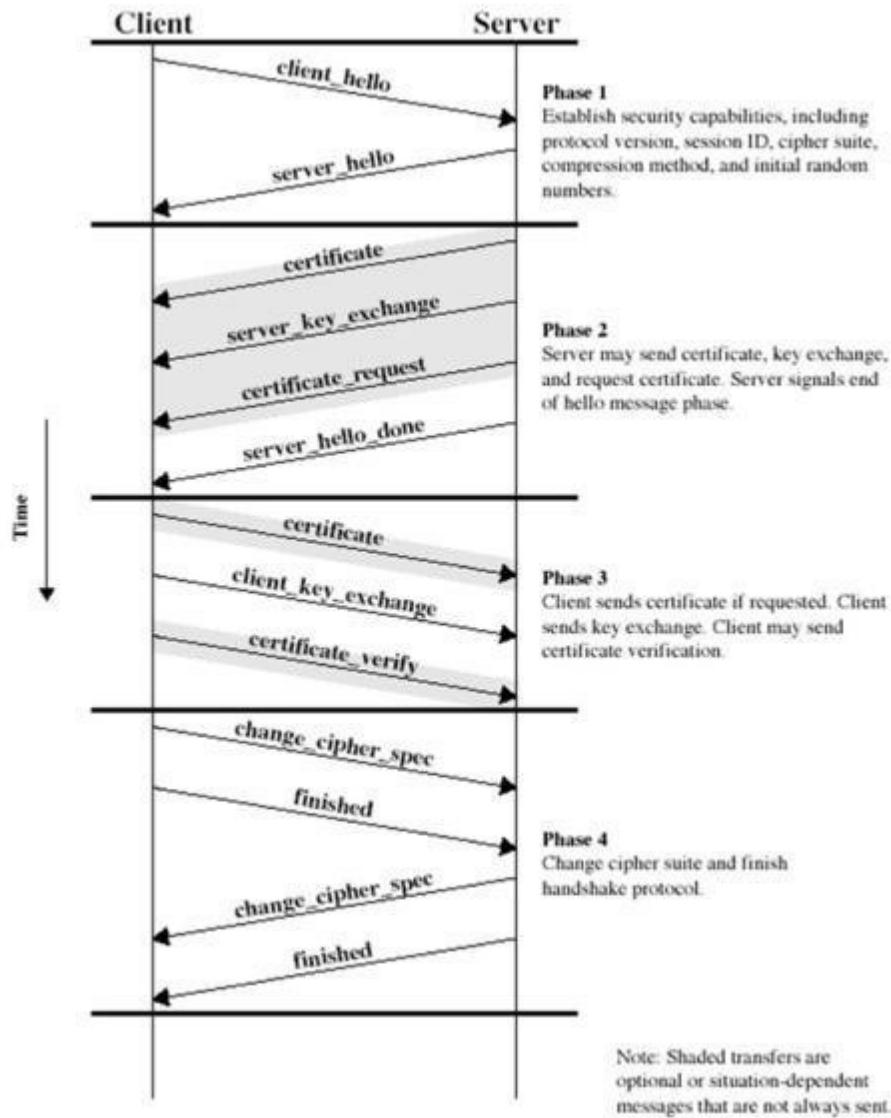

Fig 5.7 Four phases of SSL handshake protocol

SSL Record Protocol
- This protocol provides two services for SSL connections:
1. Confidentiality - using conventional encryption.
2. Message Integrity - using a Message Authentication Code (MAC).

   IIn order to operate on data the protocol performs the following
- It takes an application message to be transmitted and fragments it into manageable blocks. These blocks are 214 = 16, 384 bytes or less.
- These blocks are then optionally compressed which must be lossless and may not increase the content length by more than 1024 bytes.
- A message authentication code is then computed over the compressed data using a shared secret key. This is then appended to the compressed (or plaintext) block.

- The compressed message plus MAC are then encrypted using symmetric encryption. Encryption may not increase the content length by more than 1024 bytes, so that the total length may not exceed 214 + 2048. A number of different encryption algorithms are permitted.
- The final step is to prepend a header, consisting of the following fields:
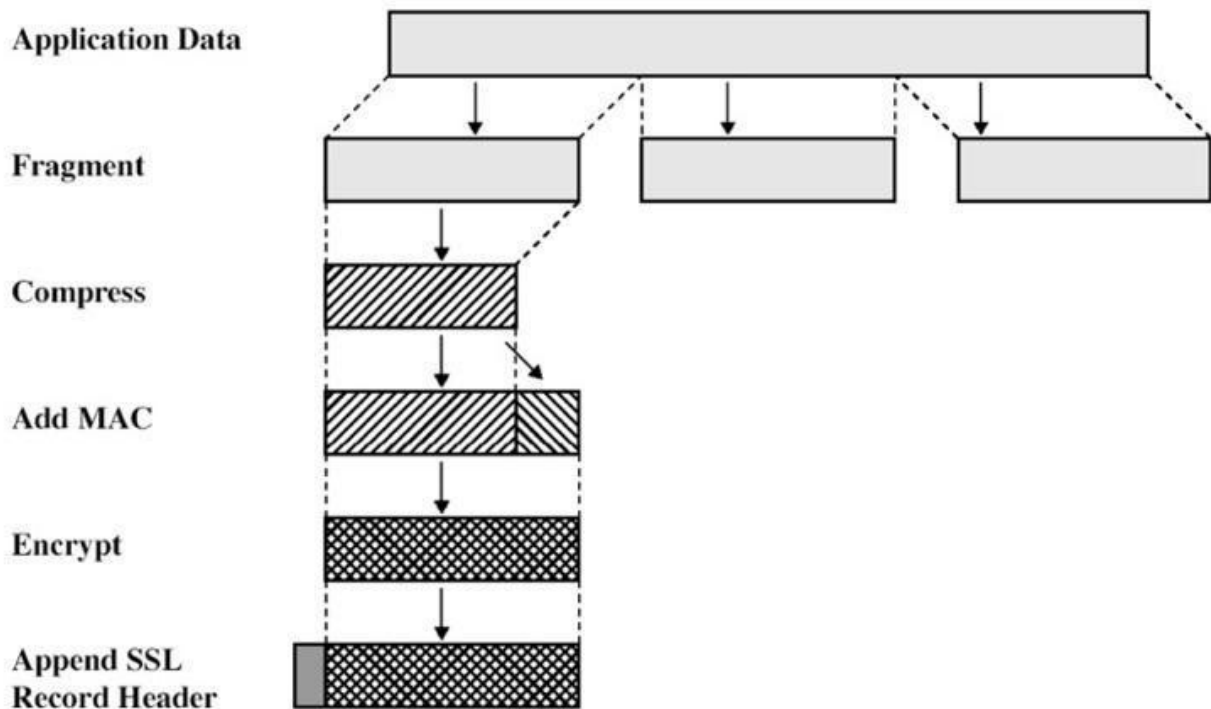
Fig 5.8 SSL Record Protocol

2. **SSL Record Protocol Operation**.

- Content type (8 bits) - The higher layer protocol used to process the enclosed fragment.
- Major Version (8 bits) - Indicates major version of SSL in use. For SSLv3,The value is 3.
- Minor Version (8 bits) - Indicates minor version in use. For SSLv3, the value is 0.
- Compressed Length (16 bits) - The length in bytes of the compressed (orplaintext) fragment.
- The "content type" above is one of four types; the three higher level protocols given above that make use of the SSL record, and a fourth known as "application data". The first three are described next as they are SSL specific protocols.

Alert Protocol

This protocol is used to convey SSL-related alerts to the peer entity. It consists of two bytes the first of which takes the values 1 (warning) or 2 (fatal). If the level is fatal SSL immediately terminates the connection. The second byte contains a code that indicates

5.3 TRANSPORT LAYER SECURITY

- The SSL protocol was originally developed at Netscape to enable ecommerce transaction security on the Web, which required encryption to protect customers' personal data, as well as authentication and integrity guarantees to ensure a safe transaction.

- To achieve this, the SSL protocol was implemented at the application layer, directly on top of TCP , enabling protocols above it (HTTP, email, instant messaging, and many others) to operate unchanged while providing communication security when communicating across the network.

- When SSL is used correctly, a third-party observer can only infer the connection endpoints, type of encryption, as well as the frequency and an approximate amount of data sent, but cannot read or modify any of the actual data.
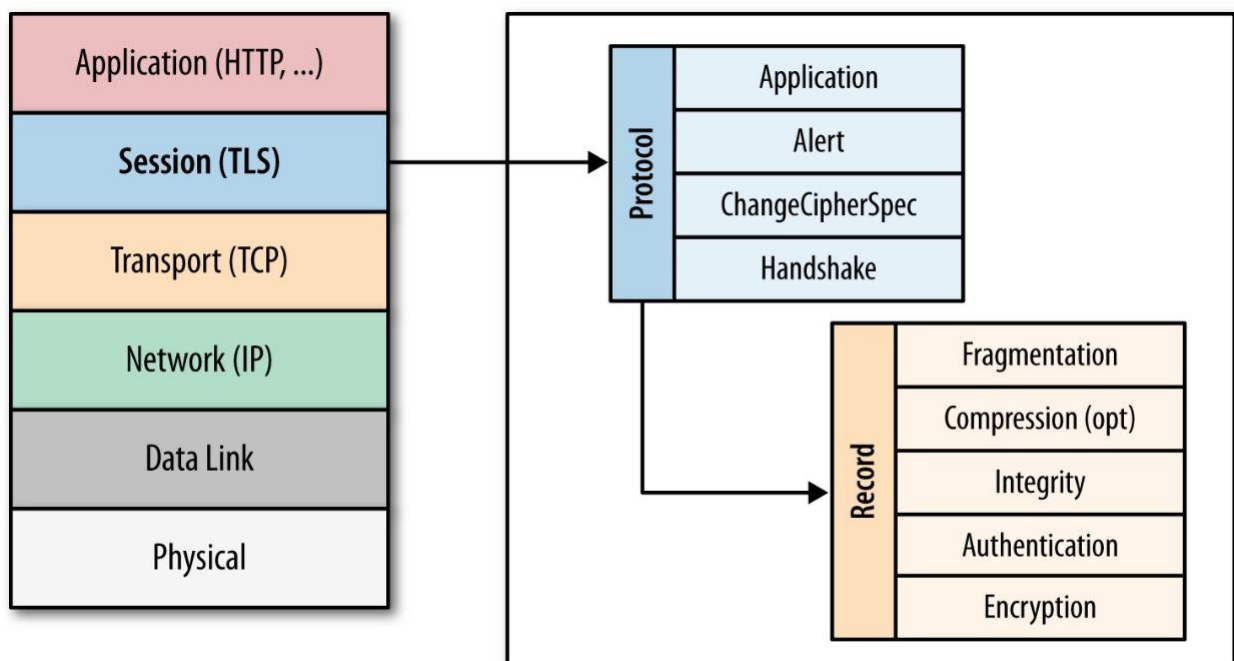
Fig 5.9Transport Layer Security (TLS)

- When the SSL protocol was standardized by the IETF, it was renamed to Transport Layer Security (TLS). Many use the TLS and SSL names interchangeably, but technically, they are different, since each describes a different version of the protocol

DIFFERENCE BETWEEN SSL AND TLS

- The differences between the two protocols are very minor and technical, but they **are** different standards. TLS uses stronger encryption algorithms and has the ability to work on different ports. Additionally, TLS version 1.0 does not interoperate with SSL version 3.0.

- Netscape originally developed the <u>SSL</u> (Secure Sockets Layer) protocol to transmit information privately, ensure message integrity, and guarantee the server identity. SSL works mainly through using public/private key encryption on data.

- It is commonly used on web browsers, but SSL can also be used with email servers or any kind of client-server transaction. For example, some instant messaging servers use SSL to protect conversations.

- The Internet Engineering Task Force (IETF) created TLS (Transport Layer Security) as the successor to SSL. It is most often used as a setting in email programs, but, like SSL, TLS can have a role in any client-server transaction.

5.4 SECURE HYPER TEXT TRANSFER PROTOCOL(SHTTP)

- Protocol (HTTP) that allows the secure exchange of files on the World Wide Web. Each S-HTTP file is either encrypted, contains a digital certificate, or both.

- For a given document, S-HTTP is an alternative to another well-known security protocol, Secure Sockets Layer (SSL). A major difference is that S-HTTP allows the client to send a certificate to authenticate the user whereas, using SSL, only the server can be authenticated.

- S-HTTP is more likely to be used in situations where the server represents a bank and requires authentication from the user that is more secure than a userid and password.

- S-HTTP does not use any single encryption system, but it does support the Rivest-Shamir-Adleman public key infrastructure encryption system. SSL works at a program layer slightly higher than the Transmission Control Protocol (TCP) level. S-HTTP works at the even higher level of the HTTP application. Both security protocols can be used by a browser user, but only one can be used with a given document. Terisa Systems includes both SSL and S-HTTP in their Internet security tool kits.

5.5 <u>TIME STAMPING PROTOCOL (TSP)</u>
- The time stamping protocol (TSP) provides proof that a certain piece of data existed at a particular time .
- This service is provided by an authority called as Time stamping authority (TSA). TSP is currently under the development of the PKIX working group.
- Using the time stamping technique ,we can ascertain whether an electronic document was created or signed at or before a particular date and time .
- This can have serious legal implications, now that digital signatures are almost as good as pen-and-paper signatures. The TSA acts like a trusted third –party notary in this scheme.

  Step1:Message digest calculation : firstly, the entity (client) requiring a time stamp calculates a message a digest of the original message, which needs a timestamp from the

TSA. The client should use a standard message digest algorithm, such as MD5 or SHA-1 for this purpose.

Step 2 Time stamping request : Now ,the client sends the message digest calculated in step 1 to the time stamp Authority (TSA) for getting it time stamped. This is called as a time stamping request.

Step 3 Time stamping response: In response to the client"s request , the TSA might decide to grant or reject the time stamp. If it decides to accept the request and process it , it signs the client"s request together with the time stamp by the TSA private key. Regardless, it returns a time stamping response back to the client.

5.6 <u>SECURE ELECTRONIC TRANSACTION (SET)</u>

- The secure electronic transaction (SET) protocol is the protocol used to facilitate the secure transmission of consumer credit card information over insecure networks, such as the Internet.
- SET blocks out the details of credit card information, thus preventing merchants, hackers and electronic thieves from accessing this information.
- SET was developed by SETco, led by VISA and MasterCard starting in 1996.
- SET was based on X.509 certificates with several extensions. The first version was finalised in May 1997 and a pilot test was announced in July 1998.
- SET makes use of Netscape's Secure Sockets Layer (SSL), Microsoft's Secure Transaction Technology (STT), and Secure Hypertext Transfer Protocol (S-HTTP). SET uses some but not all aspects of a public key infrastructure (PKI).SET allowed parties to identify themselves to each other and exchange information securely.
- SET used a cryptographic blinding algorithm that, in effect, would have let merchants substitute a certificate for a user's credit-card number.

<u>SET Participants</u>

- The following are the participants in the SET system:

  o Cardholder Acquirer
  o Merchant
  o Issuer o
  Acquirer
  o Certificate Authority

Cardholder

- In the electronic environment, consumers and corporate purchasers interact with merchants from personal computers over the Internet. A cardholder is an authorized holder of a payment card that has been issued by an issuer.

Merchant

- A merchant is a person or organization that has goods and services to sell to the cardholder. Typically, these goods and services are offered via a Web site or by electronic mail. A merchant that accepts payment cards must have a relationship with an acquirer.

Issuer

- This is a financial institution, such as a bank, that provides the cardholder with the payment card.

Acquirer

- This is a financial institution that establishes an account with a merchant and processes payment card authorizations and payments. Merchants will usually accept more that one credit card brand but do not want to deal with multiple bankcard associations or with multiple individual issuers. The acquirer provides authorization to the merchant that a given card account is active and that the proposed purchase does not exceed the credit limit. The acquirer also provides electronic transfer of payments to the merchant"s account.

Certification Authority (CA)

- This is an entity that is trusted to issue X509v3 <u>public-key certificates</u> for cardholders, merchants, and payment gateways. The success of SET will depend on the existence of a CA infrastructure available for this purpose.

THE SET PROCESS :-
1. **The customer opens an account** – the customer opens a credit card account (such as master card or visa) with a bank (issuer) that supports electronic payment mechanisms and the SET protocol.
2. **The customer receives a certificate** –After the customer"s identity is verified (with the help of details such as passport,business documents etc .) , the customer receives a digital certificate from a CA.The certificate also contains details such as the customer"s public key and its expiration date .
3. **The merchant receives a certificate-** A merchant that wants to accept a certain brand of credit cards must possess a digital certificate .
4. **The customer places an order-**this is a typical shopping card process where in the customer browses the list of items available, searches as for specific items , selects one or more of them and places the order.The merchant, in turn , sends back details such as the list of item selected,their quantities , prices, total bill, etc . Back to the customer for his record , with the help of an order form.
5. **The merchant is verified.** In addition to the order form, the merchant sends a copy of its certificate, so that the customer can verify that he or she is dealing with a valid store.
6. **The order and payment is verified.** The customer sends both order and payment information to the merchant, along with the customer"s certificate. The order confirms the purchase of the items in the order form. The payment contains credit card details. The payment information is encrypted in such a way that it cannot be read by the merchant. The customer"s certificate enables the merchant to verify the customer.
7. **The merchant requests payment authorization.** The merchant sends the payment information to the payment gateway, requesting authorization that the customer"s available credit is sufficient for this purchase.
8. **The merchant confirm the order.** The merchant sends confirmation of the order to the customer.
9. **The merchant provides the goods or service.** The merchant ships the goods or provides the service to the customer.

10. **The merchant request payment.** This request is sent to the payment gateway, which handles all of the payment processing.

SET Internals

The major transaction supported by SET are
1. Purchase Request
2. Payment Authorization
3. Payment Capture

1 Purchase Request

- The purchase request exchange consists of four messages:

- Initiate Request
- Initiate Response
- Purchase Request
- Purchase Response

Initiate Request

- To send SET messages to the merchant, the cardholder must have a copy of the certificates of the merchant and the payment gateway.
- The customer requests the certificates in the Initiate Request message, sent to the merchant. This message includes the brand of the credit card that the customer is using the message also includes an ID assigned to this request/response pair by the customer.

Initiate Response

- The merchant generates a response and signs it with its private key.
- The response includes a transaction ID for this purchase transaction. In addition to the signed response, the Initiate Response message includes the merchant's certificate and the payment gateway's certificate.

Purchase Request

- The cardholder verifies the merchant and gateway certificates by means of their respective CA signatures and then creates the order information (OI) and payment information (PI). The transaction ID assigned by the merchant is placed in both the OI and PI.
- The OI doesn't contain explicit order data such as the number and price of items. Rather, it contains an order reference generated in the exchange between merchant and customer during the shopping phase before the first SET message.
- Next, the cardholder prepares the Purchase Request message .For this purpose, the cardholder generates a one-time DES encryption key, known as a *session key*. The message includes the following:

Purchase-related information.

This information will be forwarded to the payment gateway by the merchant and consists of the PI and a dual signature. The dual signature is a signature that covers both the PI and the

OI. It's constructed in such a way that both the merchant and the payment gateway can verify the signature, even though the merchant only sees the OI and the payment gateway only sees the PI. Both the PI and the dual signature are encrypted using the one-time session key. Finally, the session key is encrypted with the public key of the payment gateway and added to the message; only the payment gateway will be able to decrypt and read the session key and therefore only the payment gateway will be able to recover the PI.

- Cardholder"s Purchase Request will be forwarded to the

- Payment gateway by the merchant consisting of the PI
- The dual signature, calculated over the PI and OI, signed with the customer"s private Signature key
- the OI message digest (OIMD)
- **Order-related information.** This information is needed by the merchant and consists of the OI and the dual signature. The merchant uses the dual signature to verify that the OI is valid.
- **Cardholder certificate.** This contains the cardholder's public key. It's needed by both the merchant and the payment gateway.

Purchase Response

- When the merchant receives the Purchase Request message, it performs the following actions:
1. Verifies the cardholder certificates by means of its CA signatures.
2. Verifies the dual signature using the customer's public signature key. This ensures that the order has not been tampered with in transit and that it was signed using the cardholder's private signature key.

- Processes the order and forwards the payment information to the payment gateway for authorization.
- Sends a purchase response to the cardholder.

Purchase Request
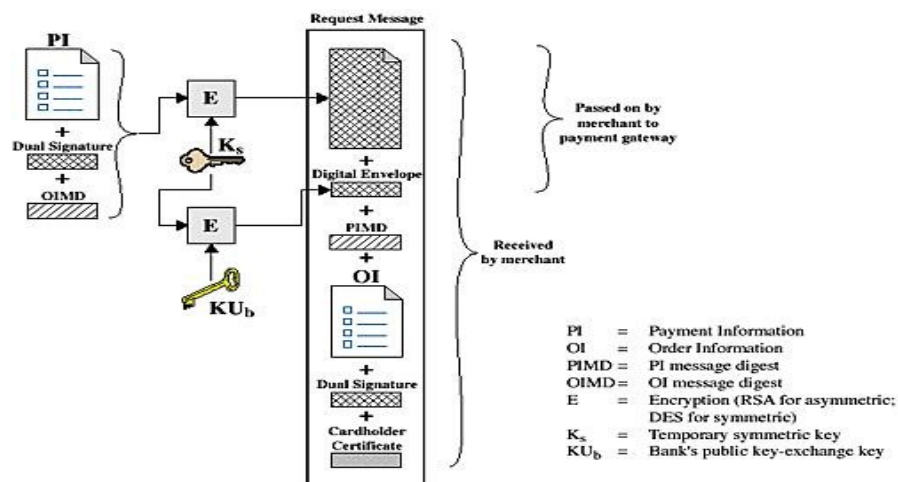
Fig 5.10 Purchase Request

2 Payment Authorization

- During the processing of an order from a cardholder, the merchant authorizes the transaction with the payment gateway.
- The payment authorization ensures that the transaction was approved by the issuer. This authorization guarantees that the merchant will receive payment, the merchant can therefore provide the services or goods to the customer. The payment authorization exchange consists of two messages: Authorization Request and Authorization response. The merchant sends an **Authorization Request** message to the payment gateway, which consisting Purchase Related Information, Authorization related information and certificates.

Authorization Request

a. **Purchase-Related information.** This information was obtained from the customer and consists of:

   The PI

- The dual signature, calculated over he PI and OI, signed with the customer"s private signature key
- The OI message digest (OIMD)

   The digital envelope

b. **Authorization-related information**.

   This information is generated by the merchantand consists of

- An authorization block that includes the transaction ID, signed with the merchant"s private signature key and encrypted with a one-time symmetric key generated by the merchant
- Digital envelope. This is formed by encrypting the one-time key with the payment gateway"s public key-exchange key.

c. **Certificates.**

   The merchant includes the cardholder"s signature key certificate (used to verify the dual signature), the merchant"s signature key certificate (needed in the payment gateway"s response).

   The payment gateway performs the following tasks:

   1. Verifies all certificates

   2. Decrypts the digital

   3. Verifies the merchant"s signature on the authorization block

   4. Decrypts the digital envelope of the payment block to obtain the symmetric key and then decrypts the payment block

   5. Verifies the dual signature on the payment block

   6. Verifies that the transaction ID received from the merchant matches that in the PI received (indirectly) from the customer

   7. Requests and receives an authorization from the issuer

   Having obtained authorization from the issuer, the payment gateway returns an

Authorization Response

   message to the merchant. It includes the following elements:

   1. Authorization- related information. Includes an authorization block, signed with the gateway"s private signature key and encrypted with a one-time symmetric key generated by the gateway. Also includes a digital envelope that contains one-time key encrypted with the merchant public key-exchange key.

   2. Capture token information. This information will be used to effect payment later.This block is of the same form as (1)-namely, assigned, encrypted capture token together with a digital

envelope. This token is not processed by the merchant. Rather,it must be returned, as is, with a payment request.

3. Certificate. The gateway"s signature key certificate. With the authorization from the gateway, the merchant can provide the goods orservice to the customer.

3   Payment Capture

- To obtain payment, the merchant engages the payment gateway in a payment capturetransaction, consisting of a capture request and a capture response message.

- For the **Capture Request** message, the merchant generates, signs, and encrypts a capture request block, which includes the payment amount and the transaction ID. The message also includes the encrypted capture token received earlier for this transaction, as well as the merchant"s signature key and key-exchange key certificates.

   - When the payment gateway receives the capture request message, it decrypts and verifies the capture request block and decrypts and verifies the capture token block. It then checks for consistency between the capture request and capture token. It then creates a clearing request that is sent to the issuer over the private payment network.

   This request causes funds to be transferred to the merchant"s account.The gateway then notifies the merchant of payment in a **Capture Response message.**

- The message includes a capture response block that the gateway signs and encrypts.

   The message also includes the gateway"s signature key certificate. The merchant software stores the capture response to be used for reconciliation with payment received from the acquirer

USER AUTHENTICATION

User authentication

6.1 Authentication basics

- Authentication means verifying the identity of someone (a user, device, or an entity) who wants to access data, resources, or applications. Validating that identity establishes a trust relationship for further interactions.

- Authentication also enables accountability by making it possible to link access and actions to specific identities. After authentication, authorization processes can allow or limit the levels of access and action permitted to that entity.

6.2 Password

Passwords are the most common form of authentication. A password is a string of alphabets, number and special character, which is supposed to be known only to the entity that is bing authenticated.

Simple password authentication offers an easy way of authenticating users. In password authentication, the user must supply a password for each server, and the administrator must keep track of the name and password for each user, typically on separate servers.

Steps in Password-Based Authentication

- Every user in the system is assigned a user id and initial password. The password is stored in clear text. in the user database against the user id. Figure shows the steps involved in authenticating a client by using a user id (name) and password.
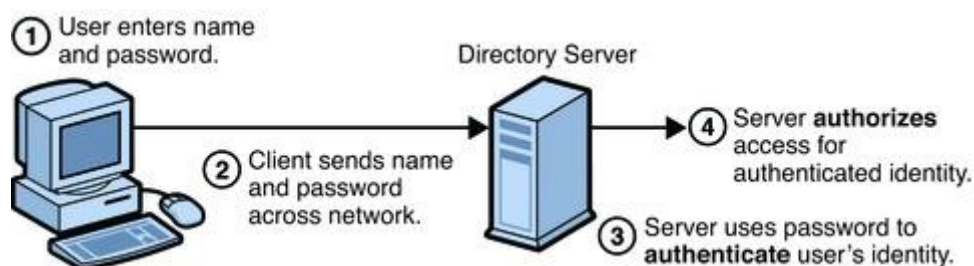
Fig 6.1 Password-Based Authentication

- In Fig 6.1 , password authentication is performed in the following steps.

1. **Prompt For user id and Password-** The application sends a screen to the user, prompting user id and password.
2. **User enters user id and password-** The user enters her id and password and presses the ok button.This causes the user id and password to travel in clear text to the server.
3. **User id and pass word validation-** the server consults the user database to validate the user id and password. This job is done by user authentication program.
4. **Authentication Results-**Depending on the failure or success of the validation of the user id and password, the user authentication program return an appropriate result back to the server.
5. **Inform user accordingly-**Depending on the outcome (success or failure), the server sends back an appropriate screen to the user. If the user authentication is successful, the serve rsends a menu option, which lists the action user is allowed to perform. If the user authentication is failure, the server sends an error screen to the user.

   Problem with the scheme

- Database contains passwords in clear text.
- Passwords travels in clear text from the user computer to the server.

Something derived from passwords:-

- The variation from the basic password based authentication is not to use the password itself, but to use something that is derived from the password.
- That is, instead of storing the password as it is, or in an encrypted format, we can run some algorithm on the password and store the output of this algorithm as he (derived) password in the database.
- When the user wants to authenticate, the user enters the password and the user"s computer performs the same algorithm locally, and sends the derived password to the server, where it is verified.
- Several requirements needs to be met to ensure that this scheme works correctly:-

  ☁ Each time the algorithm is executed for the same password, it must produce the same output.

  ☁ The output of the algorithm (i.e. something derived from the password) must not provide any clues regarding the original password.

  ☁ It should be infeasible for an attacker to provide an incorrect password and yet obtain the correct derived password.

- **Message digest for passwords:-**a simple technology to avoid the store age and transmission of clear text passwords is the use of message digests .let us understand how it works.
- **Step1:-storing message digests as derived passwords in the user database** rather than storing passwords, we can store the message digests of the passwords in the database.
- **Step2:-user authentication** when a user needs to be authenticated , the user enter the id an password , as usual, now the user"s computer computes the message digests

oh the password and send s the user id and message digest of password to the server for authentication.

- **Step 3:- server side validation** the user id and the message digest of the password travel to the server over the communication link. The server passes these values to the user authentication program which validates the user id and message digest of the password against the database and returns an appropriate response back to the server, the server uses the result if the operation to return an appropriate message back to the user.Is this approach of using the message digests of passwords completely secure , then? Let us review our original requirements:-

☁ Each time the algorithm is executed for the same password, it must produce the same output.

☁ The output of the algorithm (i.e. something derived from the password) must not provide any clues regarding the original password.

☁ It should be infeasible for an attacker to provide an incorrect password and yet obtain the correct derived password.

- The attackers can simply copy the user id and the message digest of the password, and submit them after some time to the same server as a part of a new login request .the server has no way of knowing that this login attempt as not from a legitimate user, but that it is actually from an attacker. therefore, the server would authenticate the attacker successfully! This is called as a replay attack , because the attacker simply replace the sequence of the actions of a normal user

Adding Randomness:-

- To improve the security of the solution, we need to add a bit of unpredictability or randomness to the earlier scheme. This is to ensure that although the message digest of the password is always the same, the exchange of information between the user"s computer and the server is never the same. This will ensure that a replay attack is foiled. This can be achieved by using a simple technique .

- **Step 1:- storing message digests as derived passwords in the user database:-** We simply store the message digests of the user passwords, and not the passwords themselves, in our user database.

- **Step 2: user sends a login request:-** this is an intermediate step, user login processes.` Here the user sends the login request only with her user id and neither the password, nor the message digest of the password. We shall use this concept on many occasions. As we progress further, we will notice that this results into two different login requests from the user to the server, he first one containing only the user id, and the second one containing some additional information.

- **Step 3: server crates a random challenge:-** when the server receives the user"s login request containing the user id alone, it first checks to see if the user id is a valid one(note that only the user id is checked). If it is not, it sends an appropriate error message back to the user. If the user id is valid, the server now creates a **random challenge(** a random number, generated using a pseudo-random number generation technique) and sends it back to the user. The random challenge can travel as plain text from, the server to the user"s computer.

- **Step 4:user signs the random challenge with the message digest of the password:-** at this stage the application displays the password entry screen to the user. In response, the user enters the password on the screen. The application executes the appropriate message digest algorithm on the user"s computer to create a message digest of the

password entered by the user. This message digest of the password is now used to encrypt the random challenge received from the server. This encryption is of course, f a symmetric key encrypting form.

- **Step 5: server verifies the encrypted random challenge received from the user:-** the server receives the random challenge, which was encrypted by the password of the user"s message digest. In order to verify that the challenge was indeed encrypted by the password of the user"s message digest.

- The server can decrypt the encrypted random challenge received from the user with the message digest of the user"s password. As we know, the message digest of the user"s password is available to the server via the user database. If this decryption matches the original random challenge available on the server, the server can be assured that the random challenge was indeed encrypted by the message digest of the password of the user.

- Alternatively the server can simply encrypt its own version of the random challenge(i.e the one which was sent earlier to the user) with the message digest of the user"s password. If this encryption produces an encrypted random challenge, which matches with the encypted random challenge received from the user, the server can be assured that the random challenge was indeed encrypted by the message digest of the user"s password.

- **Step 6 server returns an appropriate message back to the user:-** finally, the server sends an appropriate message back to the user, depending on whether the previous operation yielded success or failure. Note that the random challenge is different every time. Therefore, the random challenge encrypted with the message digest of the password would also be different every time. Therefore, an attacker attempting a replay attack is quiet unlikely to succeed now. This is the basis for many real-life authentication mechanisms, including Microsoft windows NT 4.0 uses the MD4 message digest algorithm to produce the message digests of the passwords and uses 16-bit random numbers as the random challenges.

Password encryption

Tthe transmission of clear text password, we encrypt the password on the user computer and the send it to the server for authentication. This means that we must provide for some sort of cryptographic functionality on the users computers (i.e. the client side).

- We have two encryption processes:-
  - The first encryption happens before a password is sorted in the user Database

  - The other encryption is performed on the user"s computer to encrypt the password before it is transmitted to the server.

- These two encryption operations are no way directly related to the each other .they may even be using totally different approaches to encryption(for example ,the user computer would use the symmetric key shared between the user and the server first for encryption and then the SSL session key for secure transmission, where as the server could only use the shared symmetric key ,as it does not have to perform any transmission)

- Therefore , the encrypted password in the database would not actually be same as the encrypted password coming from the user"s computer . However, the main idea here is that both the encrypted passwords – neither of them is in clear text.

- The fact that the encrypted versions of these two passwords may not be the same and that the server – side application logic would perform the necessary conversions between the two for verification is a minor technical variation, which we shall ignore.

6.3 <u>Authentication Tokens</u>

- An authentication token is an extremely useful alternative to a password. An authentication token is a small device that generates a new random value every time it is used.
- This random value becomes the basis for authentication. The small devices are typically of the size of small key chains, calculators of credit cards. Usually an authentication token has the following features:
- Processor
- Liquid crystal display(LCD)for display outputs
- Battery
- (optionally)a small keypad for entering
- information (optionally)a real-time clock
- Each authentication token (i.e. device) is pre-programmed with a unique number, called as random seed, or just seed. The seed forms the basis for ensuring the uniqueness of the output produced by the token.

Step 1: creation of a token

- Whenever an authentication token is created, the corresponding random seed is generated for the token by the authentication server (a special server that is configured to work with authentication tokens).this set is stored or pre-programmed inside token, as well as its entry is made against that user"s record in the user database.
- Conceptually this seed is as the user"s password (although this is technically completely different from a password), the user does not known about the value of the seed, unlike a password. This is because the seed is used automatically by the authentication token.

Step 2: use of token

- An authentication token automatically generates pseudorandom numbers, called is one-time password .or one-time passwcodes are generated randomly by an authentication token, based on the seed value that they are pre-programed with.
- They are one time because they are generated, used once, and discarded forever. When a user wants to be authenticated, the user will get a screen to enter the user id and the latest one time password .for this, the user will wants enter the user id and the one time password obtained from the authentication token.
- The user id and password travel to the server as a part of the loginrequest. The server obtains theseed corresponding in the user id from the user database, asseed retrieval program .it then calls another program called as password validation program, to which the server gives the seed and the one time password.
- If a user loses an authentication token? Can another user simply grab useit? To deal with such situation, the authentication token be generally protected by a password or a 4digit pin. Only when this PIN is entered can the one time password be generated .This also basis for what is called as multi-factor authentication.

Step 3: server returns an appropriate message back to the user

- Finally, the server sends an appropriate message back to the user, depending on whether the previous operations yielded success or failure.

Authentication token types

- There are two main types of authentication tokens.
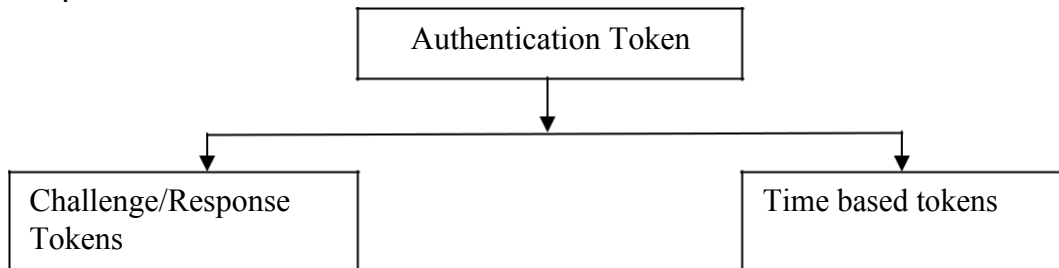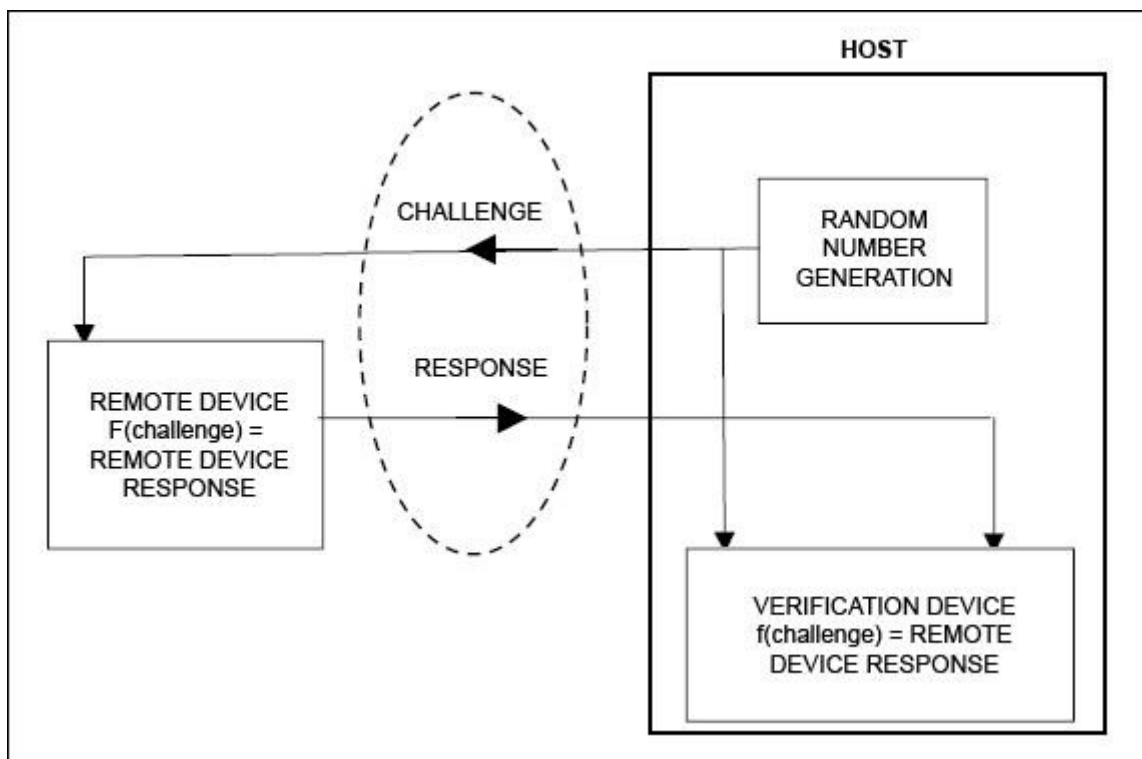1. Challenge/Response Tokens
2. Time based Tokens

.

Fig 6.2 Authentication token types

1. Challenge/response:-

- Challenge-response authentication is a family of protocols in which one party presents a question ("challenge") and another party must provide a valid answer ("response") to be authenticated. The simplest example of a challenge-response protocol is password authentication, where the challenge is asking for the password and the valid response is the correct password.

Steps Of Challenge/response works

Step1: User sends a login request

- In this technique, the user sends the login request only with her user id.

Step2: Server creates a random challenge

- When the server receives the user‚s login request containing the user id alone it first checks to see if the user id is a valid one. If the user id is valid, the server now creates random challenge (a random number, generated using a pseudo-random number generation technique) and sends it back to the user.
- The random challenge can travel as plan text from the server to the user"s computer.

Step3: User signs the random challenge with the message digest of the password

- The user gets a screen, which displays the user id, the random challenge received from the server, and data entry field, with the label password. Let us assume that the random challenge sent by the user was 8102811291012.
- At this stage, the user reads the random challenge displayed on the screen. It first opens the token using her PIN and the keys in the random challenge received from the server inside the token. In order to do this, the token has a small keypad.
- The token accepts the random challenge, and encrypt it with seed value, which is known only to itself. The result is the random challenge encrypted with the seed. This result is displayed on the display (LCD) of the token.
- The user reads this value and types it in the password field on the screen. This request is then sent to the server as the login request.

Step3: server verifies the encrypted random challenge received from the user

- The server receives the random challenge, which was encrypted with the seed by the user"s authentication token. In order to verify that the random challenge was indeed encrypted by the correct seed, the server must perform an identical operation.

- The server can decrypt the encrypted random challenge received from the user with the seed value for the user. As we know; the seed for the user is available to the server via the user database. If this decryption matches the original random challenge available on the server, the server can be assured that the random challenge was indeed encrypted by the correct seed of the user"s authentication token.

- Alternatively, the server can simply encrypt its own version of the random challenge (i.e. the one which was sent earlier to the user) with the seed for the user. If this encryption produces an encrypted random challenge, which matches with the encrypted random challenge received from the user, the server can be assured that the random challenge was indeed signed by the correct seed.

Step4: server returns an appropriate message back to the user

- Finally, the server sends an appropriate message back to the user, depending on whether the previous operations yielded success or failure.

Fig 6.3 Challenge/Response token

2. Time based tokens

- In the challenge /response mechanism discussed earlier the user has to make three entries: firstly the user has to enter the PIN to access the token; secondly, the user has to read the random challenge from the screen and key in the random number challenge into the token, and thirdly, the user has to read the encrypted random challenge from the LCD of the token and enter it into the password field.

- Users generally make quite a few mistakes in all this process, resulting into a lot of flow of wasteful information between the user"s computer, the server and the authentication token.
- In time based token the server need not send any random challenge to the user. The theory behind this is usage of time as variable input to the authentication process.

Steps of Time based tokens works

Step1: password generation and login request
- The token is pre-performed with seed usual. The copy of the seed is also available to the authentication server. As we know a challenge response token performs an operation such as encryption or message digest creation only based on the user"s inputs.However,in the case of time-based tokens.
- This is handled differently. These tokens do not required any user inputs. Instead, these tokens automatically generate a password every 60 second and display the latest password on the LCD output for the user to read and use it.
- For generation the password, the time-based tokens use two parameters: the seed and the system time. It performs some cryptographic function on these two input parameters to produce the password automatically.
- The token then displays it onto the LCD.Whenever a user wishes to log on she takes a look at the LCD display, reads the password from there and uses her id and password for login.

Step2: server-side verification
- The server receives the password .It also performs an independent cryptographic function on the user"s seed value and the current system time to generate its version of the password. If the two values match, it considers the user as a valid one.

Step3: server returns an appropriate message back to the user
- Finally, the server sends an appropriate message back to the user, depending on whether the previous operations yielded success or failure.

6.4 Certificate based authentication
- This is based on the digital certificate of a user.FIPS-196 is a standard that specifies the operation of this mechanism. As we known, in PKI, the server and (optionally) the client are required possess digital certificate in order to perform digital transaction.
- The digital certificate can then be reused for user authentication as well. In fact, if we use SSL, the server must have a digital certificate, whereas the client (user) may have digital certificates. This is because the client authentication is optional in SSL, but not the server authentication.
- Certificate based authentication is a stronger mechanism as compared to a password-based authentication mechanism, because here the user is expected to have something (certificate) and not known something (password).at the time of login.
- The user is requested to send her certificate to the server over the network as part of the login request .A copy of the certificate exists on the server, which can be used verify that the certificate is indeed a valid one.

Working of Certificate based authentication

Step 1 Creation, storage and distribution of digital certificates-

- The digital certificates are created by CA for each user and the certificate are send to the respective users. The copy of a certificate is stored by the server in its data base, in order to verify the certificate during the user"s certificate based authentication.

Step 2 Login request-

- During login request the users sends her user id to the server.

Step 3 Server creates a random challenge-

- When the server receives the user"s login request, it validate the user id. If the user id is valid, the server now creates a random number challenge and sends it back to the user.

Step 4 User signs the random challenge-

- The user has to sign the random challenge with her private key. The private key stored in a disk file on the user computer. The private key is used to encrypt the random challenge received from server to create users digital signature.
- This is done by two step first a message digest of the random challenge is created and the message digist is then encrypted with the users private key and send to the server. The server then verifies the user"s signature by obtaining the public key from the user database. The public key is used to decrypt the signed random challenge received from the user. After that it compares this decrypted random challenge with its original random challenge.

Step 3 Server returns an appropriate message back to the users

- Finally the server sends an appropriate message back to the user., wheatear the previous operation is success or failure.

6.5 Biometric authentication

- Biometrics is the science and technology of measuring and analyzing biological data. In information technology, biometrics refers to technologies that measure and analyze human body characteristics, such as DNA, fingerprints, eye retinas and irises, voice patterns, facial patterns and hand measurements, for authentication purposes.

- Authentication by biometric verification is becoming increasingly common in corporate and public security systems, consumer electronics and point of sale (POS) applications. In addition to security, the driving force behind biometric verification has been convenience.

- Biometric devices, such as finger scanners, consist of:

- A reader or scanning device

- Software that converts the scanned information into digital form and compares match points

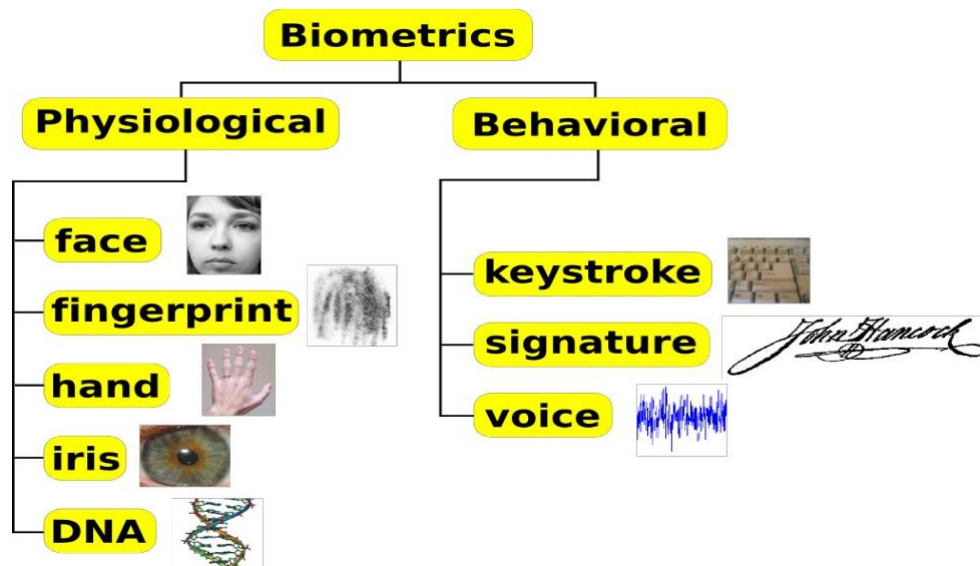- A database that stores the biometric data for comparison

Fig 6.4 Biometric Authentication

- To prevent identity theft, biometric data is usually encrypted when it's gathered. Here's how biometric verification works on the back end: To convert the biometric input, a software application is used to identify specific points of data as match points. The match points in the database are processed using an algorithm that translates that information into a numeric value. The database value is compared with the biometric input the end user has entered into the scanner and authentication is either approved or denied.

- The two basic modes of a biometric system. First, in **verification** (or authentication) mode the system performs a one-to-one comparison of a captured biometric with a specific template stored in a biometric database in order to verify the individual is the person they claim to be. Three steps are involved in the verification of a person. In the first step, reference models for all the users are generated and stored in the model database. In the second step, some samples are matched with reference models to generate the genuine and impostor scores and calculate the threshold. Third step is the testing step. This process may use a smart card, username or ID number (e.g. PIN) to indicate which template should be used for comparison. 'Positive recognition' is a common use of the verification mode, "where the aim is to prevent multiple people from using same identity".

- Second, in **identification** mode the system performs a one-to-many comparison against a biometric database in attempt to establish the identity of an unknown individual. The system will succeed in identifying the individual if the comparison of the biometric sample

to a template in the database falls within a previously set threshold. Identification mode can be used either for 'positive recognition' (so that the user does not have to provide any information about the template to be used) or for 'negative recognition' of the person "where the system establishes whether the person is who she (implicitly or explicitly) denies to be".[3] The latter function can only be achieved through biometrics since other methods of personal recognition such as passwords, PINs or keys are ineffective.

Biometric Authentications defines two configurable parameters.

- **false acceptance rate or false match rate (FAR or FMR)**: the probability that the system incorrectly matches the input pattern to a non-matching template in the database. It measures the percent of invalid inputs which are incorrectly accepted. In case of similarity scale, if the person is an imposter in reality, but the matching score is higher than the threshold, then he is treated as genuine. This increases the FAR, which thus also depends upon the threshold value.[7]
- **false rejection rate or false non-match rate (FRR or FNMR)**: the probability that the system fails to detect a match between the input pattern and a matching template in the database. It measures the percent of valid inputs which are incorrectly rejected.

Network Security & VPN
7.1 Brief introduction of TCP/IP
7.2 Firewall
7.3 IP Security
7.4 Virtual Private Network (VPN)

7.1 Brief introduction of TCP/IP

- TCP/IP protocol suite contains five main layers:
- Application Layer
- Transport Layer
- Network (or internet) Layer
- Data link Layer
- Physical Layer

Unlike the OSI protocol suite, there are no presentation and session layers in TCP/IP.The data unit initially crated at the application layer(i.e. by an application, such as email, web browser, etc)is called as a message. A message is actually broken down into segments by the transport layer. Note that the transport layer of TCP/IP contains two protocols: transmission control protocol (TCP) and user datagram protocol (UDP). The transport layer then adds its own header to the segment and gives it to the network layer. The data link layer adds the frame header and gives it to the physical layer for transmission. At the physical layer the actual bits are transmitted as voltage pulses. An opposite process happens at the destination end where each layer removes the previous layer"s header and finally the application layer receives the original message.
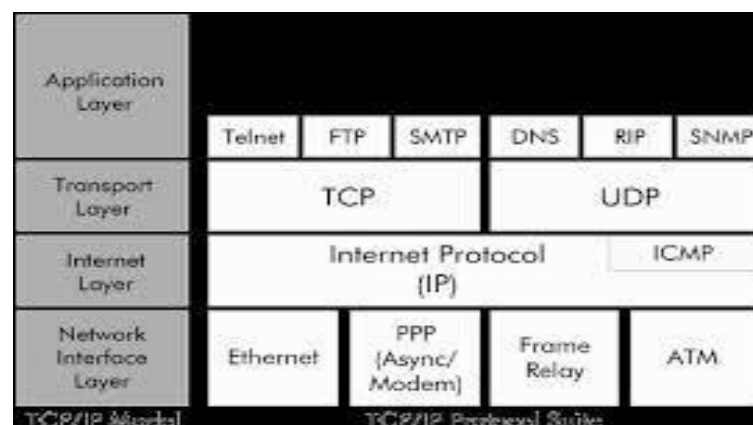
Fig 7.1 Layer in the TCP/IP protocol suite

TCP Segment Format

- The TCP , Network, and data link layer add headers to the received data block. The transport layers add header to the original message, it not only appends the header fields to the original message, but also performs some processing, such as calculating the checksum for error detection etc.
- A TCP segment consists of a header of size 20 to 60 bytes, followed by the actual data. Header fields inside the TCP Segments
- **Source port number:-** this 2-byte number signifies the port number of the source computer corresponding to the application that is sending this TCP segment.
- **Destination port number:-** this 2-byte number signifies the port number of the destination computer corresponding to the application that is expected to receive this TCP segment.
- **Sequence number:-** this 4-byte field defines the number assigned to the first byte of the data portion contained in this TCP segment. TCP is a connection-oriented protocol. For ensuring a correct delivery ,each byte is to be transmitted from the source to the destination is numbered in an increasing sequence. The sequence number field tells the destination host, which byte in this sequence comprises the first byte of the TC P segment. During the TCP connection establishment phase, both the source as well as the destination generates different unique random numbers. For instance if this random number is 3130 and the first TCP packet is carrying 2000 bytes of data, then the sequence number field for that packet would contain 3132 are used in connection establishment. The second segment would then have a sequence number of 5132(3132+2000) and so on.
- **Acknowledgement number:** if the destination host receives a segment with sequence number X correctly, it sends X+1 as the acknowledgement number back to the source. Thus this 4-byte number defines the sequence number the source is expecting from the destination as a receipt of the correct delivery.
- **Header length:** this 4- bit field specifies the number of four-byte words in the TCP header. As we know, the header length can be between 20 and 60 bytes. Therefore the value of this field can be between 5(because 5*4=20_) and 15(because 15*4=60).
- **Reserved:** this 6-bytes field is reserved for future use and is currently unused.
- **Flag:** this 6-bit field defines six different control flags, each of them occupying one bit. Out of the six flags, two are most important. The SYN flags is used when a TCP connection is being established between two hosts. Similarly, the other flag of importance is the FIN flag. If the bit corresponding to this flag is set, then that the sender wants to terminate the current TCP connection.
- **Window size:** this field determines the size of the sliding window that the other party must maintain.
- **Checksum:** this 16-bit field contains the checksum for facilitating the error detection and correction.
- **Urgent pointer:** this field is used in situation where data in a TCP segment is more important or urgent than other data in the same TCP connection.
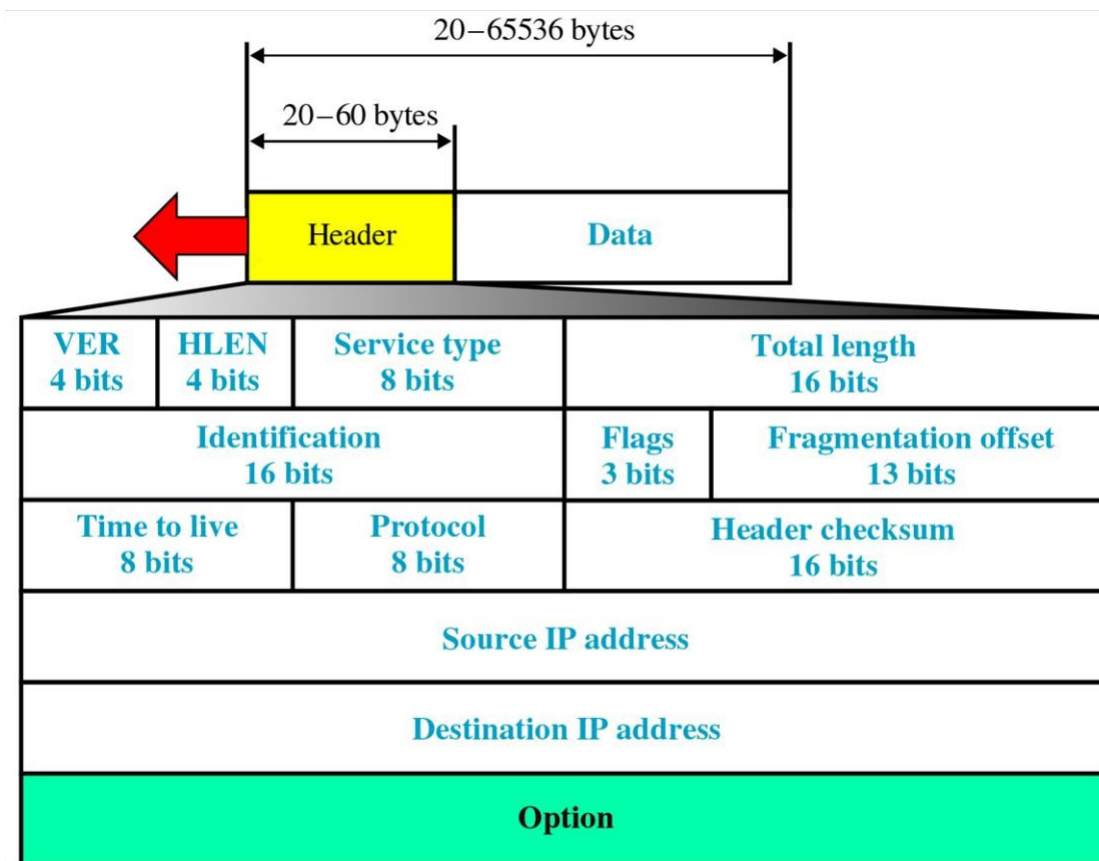
Fig 7.2 TCP Segment format

IP Datagram Format:-

The TCP header plus the original message is now passed to the IP layer. The IP layer treats this whole package of TCP header + original message as its original message and adds its own header to it. This results into the creation of an IP datagram.

- **Version: -** this field current a value 4, which indicates **IP version 4(IPv4).** In future, this fields would contain 6 when IP version 6 (IPv6) becomes the standard.
- **Header length(HLEN):-** Indicates the size of the header in a multiple of four-byte words. When the header size is 20 bytes. The value of this field is 5(because 5*4=20) and when the option field is at the maximum size, the value of HLEN is 15(because 15*4=60)
- **Service type:-** this fields is used to define service parameters such as the priority f the datagram and the level of reliability desired.
- **Total length:-** this field contains the total length of the IP datagram. Because it is two bytes long, and IP datagram cannot be more than 65,536 bytes (2^{16}=65,536).
- **Identification:-** this field is used in the situations when a datagram is fragmented. As a datagram passes through different networks, it might be fragmented into smaller sub-datagram to match the physical datagram size of the underlying network. In these situation, the sub-data grams are sequenced using the identification field, so that the original datagram can be reconstructed from them.
- **Flags:-** this field corresponds to the earlier field(identification). It indicates whether a datagram can be fragmented in the first place- and if it can be fragmented, whether it is the first or the last fragment or it can be a middle fragment , etc.

- **Fragmentation offset:-** if a datagram is fragmented, this field is useful. It is a pointer that indicates the offset of the data in the original datagram before fragmentation. This is useful when reconstructing a datagram from its fragments.
- **Time to live:-** there could be many datagram travelling in different directions through lengthy paths, trying to reach their destinations. This can create congestion and the routers may become too busy, thus bringing at least parts of the internet to a virtual halt. On some cases, the datagram can continue to travel in a loop in between without reaching the final destination and in fact, coming back to the original sender. To avoid this the datagram sender initializes this field(that is, time to live) to some number.
- **Protocol:-** this field identifies the transport protocol running on top of IP. After the datagram is constructed from its fragments it has to be passed on the upper layer software piece. This could be TCP or UDP. This field specifies which piece of software at the destination node he datagram should be passed on to.
- **Source address:-** this field contains the 32-bit IP address of the sender.
- **Destination address:-** this field contains the 32-bit IP address of the final destination.
- **Options:-** this field contains optional information such as routing details, timing, management and alignment. For instance, it can store the information about the exact route that the datagram has taken. When it passes through a router, the router puts in its id and optionally, also the time when it passed through that router, in one of the slots in this field. This helps tracing and fault detection of datagram. However most of the time, the space in this field is not sufficient for all these details, therefore, it is not used very often.
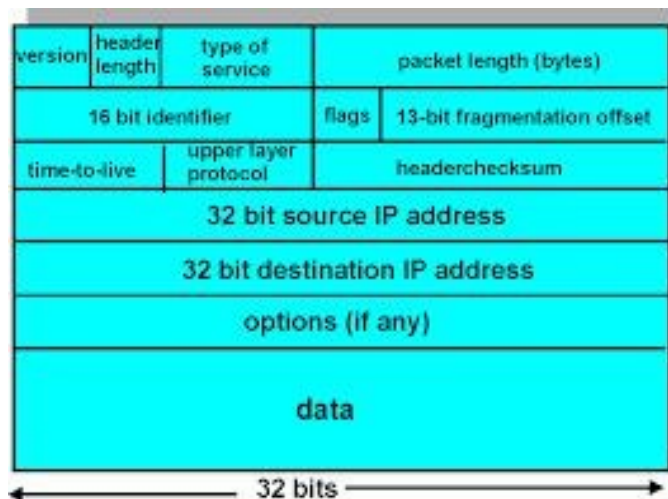
Fig 7.3 IP Datagram

7.2 Firewall

- A firewall is a system designed to prevent unauthorized access to or from a private network. Firewalls can be implemented in both hardware and software, or a combination of both. Firewalls are frequently used to prevent unauthorized Internet users from accessing private networks connected to the Internet, especially *intranets.*
- All messages entering or leaving the intranet pass through the firewall, which examines each message and blocks those that do not meet the specified security criteria. A firewall is a network security system, either hardware or software based, that controls incoming and outgoing network traffic based on a set of rules.

- A firewall is a network security system, either hardware or software based, that controls incoming and outgoing network traffic based on a set of rules. a firewall controls access to the resources of a network through a positive control model.
- This means that the only traffic allowed onto the network defined in the firewall policy is; all other traffic is denied.

Types of Firewalls

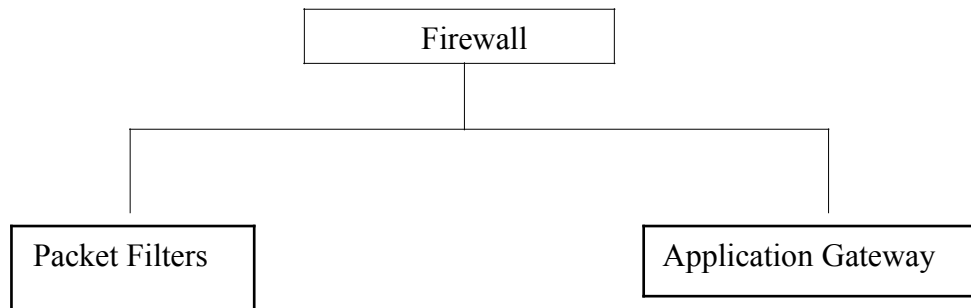- Based on the criteria for filtering traffic, firewall are generally classified into two types.

Fig 7.4 Types of Firewalls
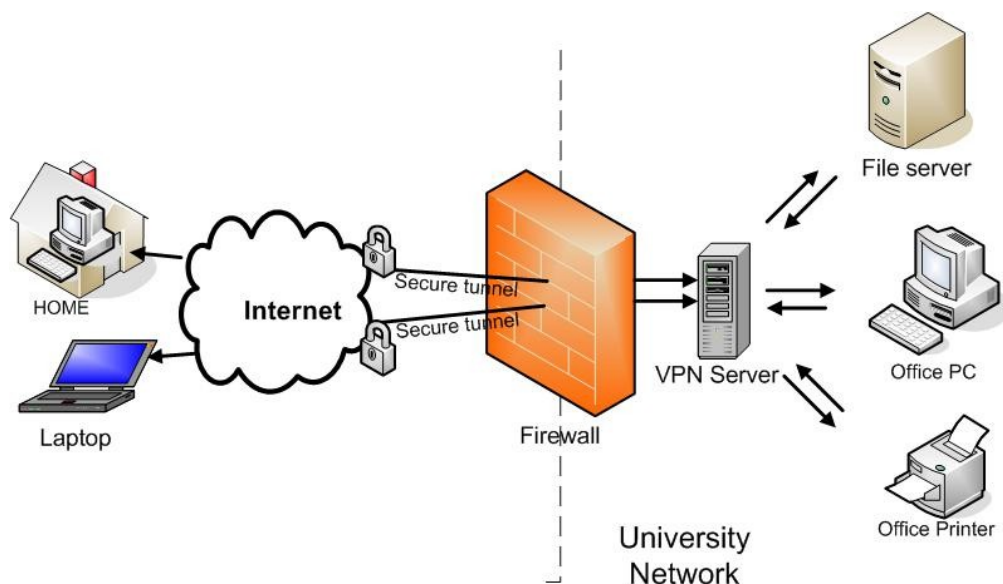
Fig 7.5 Firewall

1 Packet Filters
- Using network communication, a node transmits a packet that is filtered and matched with predefined rules and policies. Once matched, a packet is either accepted or denied. Packet filtering checks source and destination IP addresses.
- If both IP addresses match, the packet is considered secure and verified. Because the sender may use different applications and programs, packet filtering also checks source and destination protocols, such as User Datagram Protocol (UDP) and Transmission Control Protocol (TCP).
- Packet filters also verify source and destination port addresses. Some packet filters are not intelligent and unable to memorize used packets. However, other packet filters can

memorize previously used packet items, such as source and destination IP addresses. Packet filtering is usually an effective defense against attacks from computers outside a local area network (LAN). As most routing devices have integrated filtering capabilities, packet filtering is considered a standard and cost-effective means of security.

- In the context of a TCP/IP network, a packet filter watches each individual IP datagram, decodes the header information of in-bound and out-bound traffic and then either blocks the datagram from passing or allows the datagram to pass based upon the contents of the source address, destination address, source port, destination port and/or connection status. This is based upon certain criteria defined to the packet filtering tool. The leading IP routers, including Cisco, Bay, and Lucent, can be configured to filter IP datagram.
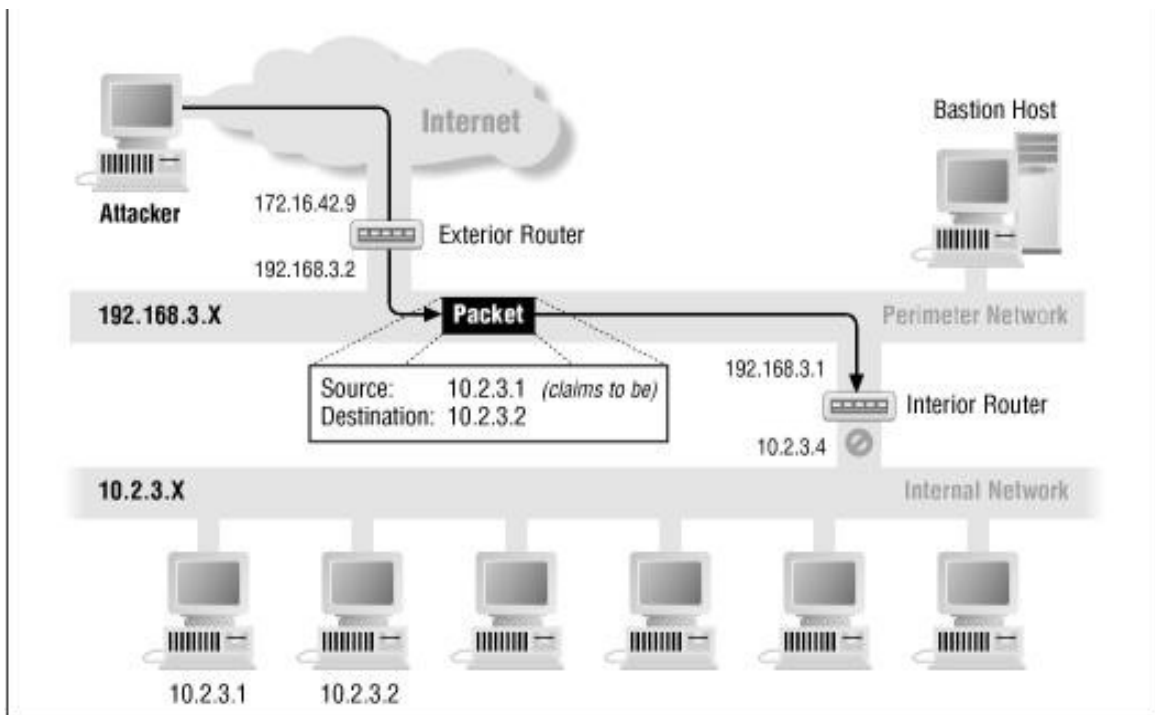
Fig 7.6 Packet Filters

How does a packet filter work?

- All packet filters function in the same general fashion. Operating at the network layer and transport layer of the TCP/IP protocol stack, every packet is examined as it enters the protocol stack. When a packet passes through a packet-filter firewall, its source and destination address, protocol, and destination port number are checked against the firewall's rule set.

- Any packets that aren't specifically allowed onto the network are dropped (i.e., not forwarded to their destination). Packet-filter firewalls work mainly on the first three layers of the OSI reference model (physical, data-link and network), although the transport layer is used to obtain the source and destination port number

- For example, if a firewall is configured with a rule to block Telnet access, then the firewall will drop packets destined for TCP port number 23, the port where a Telnet server application would be listening.

- A packet filters performs following functions

    1. Receive each packet at it arrives.
    2. Pass the packet through set of rule, based on the contents of the IP and transport header fields of the packets. if there is a match with one of the set rules, decides whether to accepts or discard the packed based on that rule.
    3. If there is no match with any rule take the packets are discarded else accepts all packets

Attackers can try and break the security of a packet filter by using following techniques.

Ip address spoofing

- Spoofing is a means to hide one"s true identity on the network. To create a spoofed identity, an attacker uses a fake source address that does not represent the actual address of the packet.
- Spoofing may be used to hide the original source of an attack or to work around network access control lists (ACLs) that are in place to limit host access based on source address rules. An intruder outside the corporate network can attempt to send a packet with IP address same as the one of the IP address of the internal users.

Prevention measures
- This attack can be defeated by discarding all the packets that arrive at the incoming side of firewall, with the source address equal to one of the internal address.

Source Routing attack
- The Source Route option in IP packets is usually used in network path troubleshooting and temporary transmission of some special services. Packets carrying the Source Route option ignore the forwarding entries of devices along the transmission path during the forwarding process.
- For example, if you want an IP packet to pass through routers R1, R2, and R3, you can specify the IP addresses of the interfaces on the three routers in the Source Route option of the packet. In this case, the IP packet will pass through R1, R2, and R3 in turn, regardless of the routing tables of the routers.
- During the transmission of an IP packet carrying the Source Route option, the source address and destination address are always changing. An attacker may forge some legal IP addresses by configuring the Source Route option, thus mingling in the target network.

Prevention measures
- Check whether received packets carry the Source Route option. If yes, drop or forward the packets and log the event, depending on the configuration.

Tiny Fragment attack

- Fragmentation is necessary in order for traffic, which is being sent across different types of network, such as Ethernet, Token Ring, X.25, Frame Relay, ATM erc. media to arrive successfully at its intended destination.
- The reason for this is that different types of network media and protocols have different rules involving the maximum size allowed for datagrams on its network segment. This is known as the maximum transmission unit or MTU.

- So in order to transmit a datagram across a network segment which has a MTU smaller than that of the packet to be transmitted fragmentation is required. In such case IP packet needs to be fragmented.An attacker use this characterstics of TCP/IP protocol. The attacker hopes that a filtering router will examine only the first fragment and allow all other fragments to pass.

Prevention measures
- This attack can be prevented at the router by enforcing rules, which govern the minimum size of the first fragment. This first fragment should be made large enough to ensure it contains all the necessary header information.

2. Application Gateways

- An application gateway is known as application proxy or application-level proxy, an application gateway is an application program that runs on a firewall system between two networks. When a client program establishes a connection to a destination service, it connects to an application gateway, or proxy.
- The client then negotiates with the proxy server in order to communicate with the destination service. In effect, the proxy establishes the connection with the destination behind the firewall and acts on behalf of the client, hiding and protecting individual computers on the network behind the firewall.
- This creates two connections: one between the client and the proxy server and one between the proxy server and the destination. Once connected, the proxy makes all packet-forwarding decisions. Since all communication is conducted through the proxy server, computers behind the firewall are protected.
- It filters incoming node traffic to certain specifications which mean that only transmitted network application data is filtered. Such network applications include File Transfer Protocol (FTP), Telnet, Real Time Streaming Protocol (RTSP) and BitTorrent.
- While this is considered a highly secure method of firewall protection, application gateways require great memory and processor resources compared to other firewall technologies, such as stateful inspection.
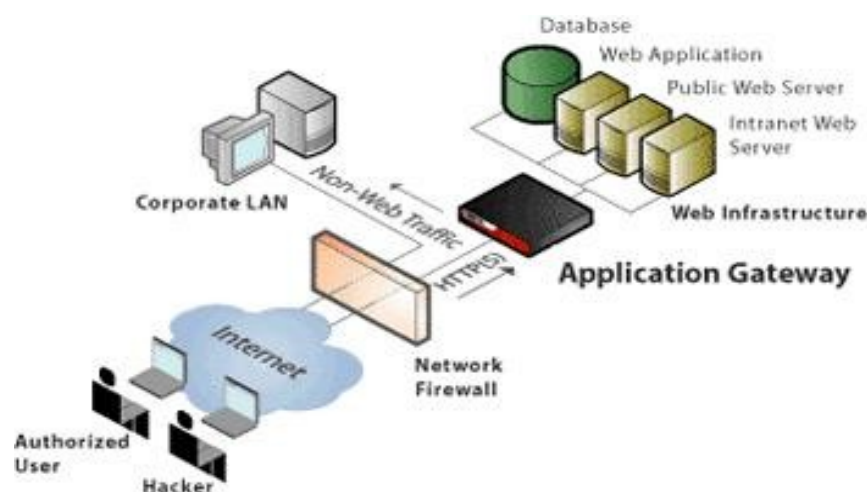

Fig 7.7 Application Gateways

Network Address Translation (NAT)

- The Internet is expanding at an exponential rate. As the amount of information and resources increases, it is becoming a requirement for even the smallest businesses and homes to connect to the Internet.
- Network Address Translation (NAT) is a method of connecting multiple computers to the Internet (or any other IP network) using one IP address. This allows home users and small businesses to connect their network to the Internet cheaply and efficiently.
- The impetus towards increasing use of NAT comes from a number of factors:
- A world shortage of IP addresses
- Security needs
- Ease and flexibility of network administration
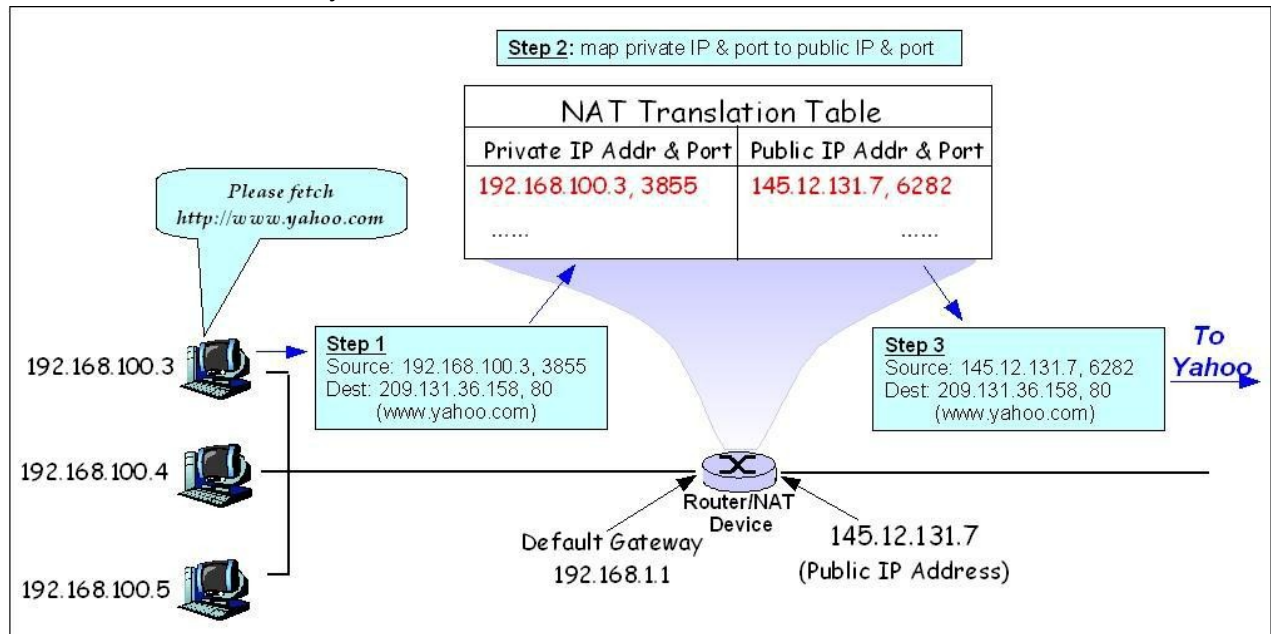
Fig 7.8 Network Address Translation (NAT)

NAT Operation

- The basic purpose of NAT is to multiplex traffic from the internal network and present it to the Internet as if it was coming from a single computer having only one IP address.The TCP/IP protocols include a multiplexing facility so that any computer can maintain multiple simultaneous connections with a remote computer.
- It is this multiplexing facility that is the key to single address NAT.To multiplex several connections to a single destination, client computers label all packets with unique "port numbers".
- Each IP packet starts with a header containing the source and destination addresses and port numbers:

Source address	Source port	Destination address	Destination port

- This combination of numbers completely defines a single TCP/IP connection. The addresses specify the two machines at each end, and the two port numbers ensure that each connection between this pair of machines can be uniquely identified.
- Each separate connection is originated from a unique source port number in the client, and all reply packets from the remote server for this connection contain the same number as their destination port, so that the client can relate them back to its correct connection.

- In this way, for example, it is possible for a web browser to ask a web server for several images at once and to know how to put all the parts of all the responses back together.
- A modern NAT gateway must change the Source address on every outgoing packet to be its single public address. It therefore also renumbers the Source Ports to be unique, so that it can keep track of each client connection.
- The NAT gateway uses a port mapping table to remember how it renumbered the ports for each client's outgoing packets. The port mapping table relates the client's real local IP address and source port plus its translated source port number to a destination address and port.
- The NAT gateway can therefore reverse the process for returning packets and route them back to the correct clients.
- When any remote server responds to an NAT client, incoming packets arriving at the NAT gateway will all have the same Destination address, but the destination Port number will be the unique Source Port number that was assigned by the NAT.
- The NAT gateway looks in its port mapping table to determine which "real" client address and port number a packet is destined for, and replaces these numbers before passing the packet on to the local client.
- This process is completely dynamic. When a packet is received from an internal client, NAT looks for the matching source address and port in the port mapping table. If the entry is not found, a new one is created, and a new mapping port allocated to the client:
    - Incoming packet received on non-NAT port
    - Look for source address, port in the mapping table
    - If found, replace source port with previously allocated mapping port
    - If not found, allocate a new mapping port
    - Replace source address with NAT address, source port with mapping port
    - Packets received on the NAT port undergo a reverse translation process:
    - Incoming packet received on NAT port
    - Look up destination port number in port mapping table
    - If found, replace destination address and port with entries from the mapping table
    - If not found, the packet is not for us and should be rejected

Firewall Configuration

- Usually, a configuration of firewall consists of more than one systems. To configure a firewall, a written security policy should beformed first. Then design a firewall to implement the policy.
- The firewall should be reviewed and updated from time to time.In the following, we discuss three common firewall configurations
1. Screened Host Firewall, Single Homed Bastion
2. Screened Host Firewall, Dual –Homed Bastion
3. screened subnet firewall

Screened Host Firewall, Single Homed Bastion

- In Screened Host Firewall, Single Homed Bastion consists of A packet filter router and an application gateway.
- The packet filter ensures that the incoming traffic (i.e. from the internet to the corporate network) is allowed only if it is destined for the application gateway, by examining the

destination address field of every incoming IP packet. Similarly, it also ensures that the outgoing traffic (i.e. from the corporate network to the internet) is allowed only if it is originating from the application gateway, by examining the source address field of every outgoing IP packet.

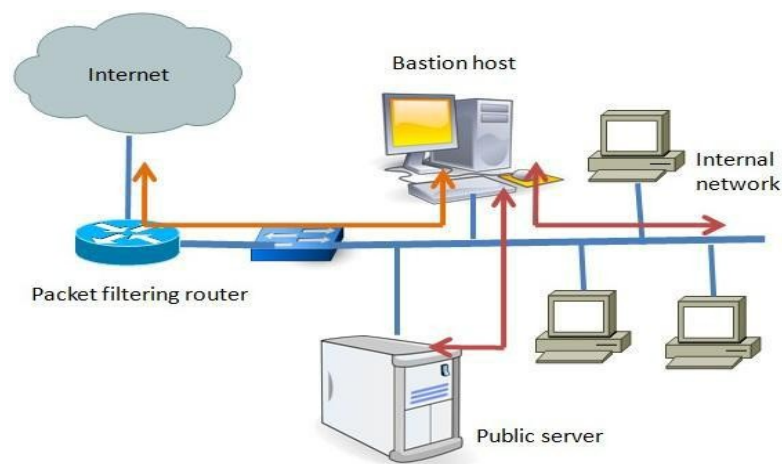☁ The application gateway performs authentication and proxy functions.

☁ In case of single-homed Bastion host, the packets come in and go out over the same network interface . So the application gateway cannot guarantee that all packets are analyzed and checked

☁ A single-homed implementation may allow a hacker to modify the router not to forward packets to the bastion host. This action would bypass the bastion host and allow the hacker directly into the network.

☁ But such a bypass usually does not happen because a network using a single-homed bastion host is normally configured to send packets only to the bastion host, and not directly to the Internet.

Fig 7.9 Screened Host Firewall, Single Homed Bastion

Screened Host Firewall, Dual –Homed Bastion

- In this case direct connections between the internal hosts and the packet filter are avoided. Instead, the packet filter connects only to the application gateway, which, in turn, has a separate connection with the internal hosts. Therefore, now even if the packet filter is successfully attacked, only the application gateway is visible to the attacker. The internal hosts are protected.
- In case of dual-homed Bastion host, the application gateway has two separate network interfaces as shown below. As a consequence, it has complete control over the packets. In this case even if the router got compromised, the internal network will remain unaffected since it is in the separate network zone. Traffic between the Internet and other hosts on the private network has to flow through the bastion host.
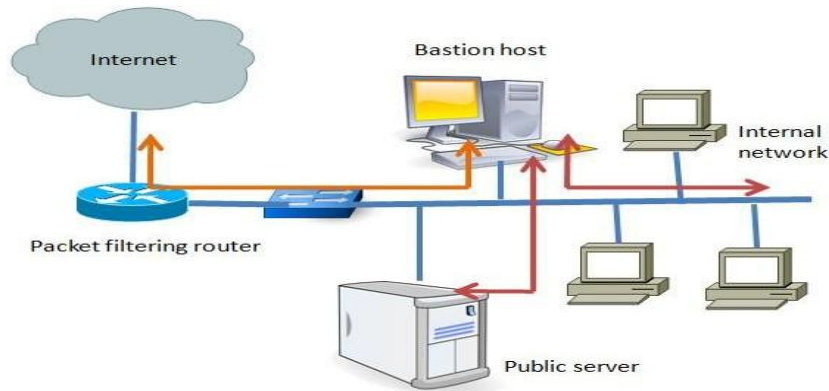
Fig 7.10 Screened Host Firewall, Dual –Homed Bastion

Screened subnet firewall

- The third configuration is screened subnet firewall. In this configuration, two packet-filtering router are used, one between the application gate way and the internet and one between the bastion host and the internal network.
- This configuration creates an isolated subnetwork, which may consist of the bastion host and/or several information services and modems for dial-in capability.
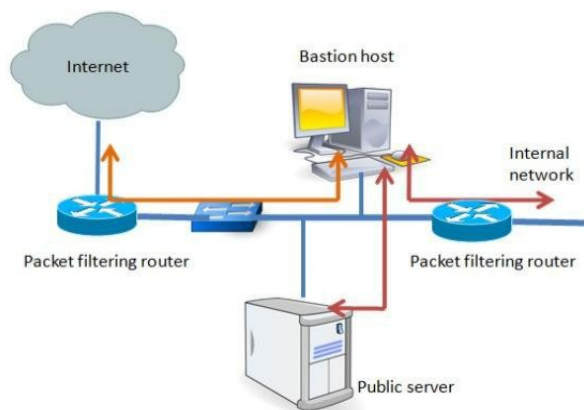
Fig 7.11 Screened subnet firewall

Demilitarized Zone

- DMZ demilitarized zone, a computer or small subnetwork that sits between a trusted internal network, such as a corporate private LAN, and an untrusted external network, such as the public Internet.
- Typically, the DMZ contains devices accessible to Internet traffic, such as Web (HTTP) servers, FTP servers, SMTP (e-mail) servers and DNS servers. In a typical DMZ configuration for a small company, a separate computer (or host in network terms) receives requests from users within the private network for access to Web sites or other companies accessible on the public network.

- The DMZ host then initiates sessions for these requests on the public network. However, the DMZ host is not able to initiate a session back into the private network. It can only forward packets that have already been requested.
- Users of the public network outside the company can access only the DMZ host. The DMZ may typically also have the company's Web pages so these could be served to the outside world. However, the DMZ provides access to no other company data. In the event that an outside user penetrated the DMZ host's security, the Web pages might be corrupted but no other company information would be exposed.

Limitation of firewell

- A firewall is a crucial component of securing your network and is designed to address the issues of data integrity or traffic authentication and confidentiality of your internal network (via NAT). Your network gains these benefits from a firewall by receiving all transmitted traffic through the firewall.
- Your network gains these benefits from a firewall by receiving all transmitted traffic through the firewall. The importance of including a firewall in your security strategy is apparent; however, firewalls do have the following limitations:
- **Insiders intrusion-** a firewall system designed to thwart outside attack, if an insider attacks I the internal network in some way; the fairewall cannot prevent such attack.
- **Direct Internet traffic-** A firewall must be configured very carefully. It is effective only if it is the only entry and exit point of an organization network. If instead the firewall is one of the entry-exit point, a user can bypass the firewall and exchange information with the internet through the other entry-exit point.
- **Virus attack-** A firewall cannot protect the internal network from virus attack.

7.3 IP Security

- IPSec protects one or more paths between a pair of hosts, a pair of security

    gateways, or a security gateway and a host. A security gateway is an intermediate device, such as a router or firewall, that implements IPsec.
    Two devices that use IPsec to protect a path between them are called peers.

IPSec Overview

- IPSec is not a single protocol, but rather a set of services and protocols that provide a complete security solution for an IP network. These services and protocols combine to provide various types of protection. Since IPsec works at the IP layer, it can provide these protections for any higher-layer TCP/IP application or protocol without the need for additional security methods, which is a major strength. Some of the kinds of protection services offered by IPsec include the following:
    - ■Encryption of user data for privacy
    - ■ Confidentiality (encryption) – ensuring that the data has not been read enroute.
    - ■Protection against certain types of security attacks, such as replay attacks.
    - ■The ability for devices to negotiate the security algorithms and keys required to meet their security needs.
    - ■Two security modes, tunnel and transport, to meet different network needs.
    - ■ Data origin authentication – identifying who sent the data.
    - ■ Connectionless integrity – ensuring the data has not been changed enroute.

       ■ Replay protection – detecting packets received more than once to help protect against denial of service attacks.

IPSEc Protocols Two primary types of IP Security (IPSec) protocols exist: Encapsulating Security Payload (ESP) and Authentication Header (AH). ESP provides authentication and encryption; AH provides authentication but not encryption. IPSec also implementation Data Encryption Standard (DES) or Triple DES (3DES) for encryption. Both AH and ESP can used one of the two mode.

1. Tunnel Mode
2. Transport Mode

1. Tunnel Mode

IPSec tunnel mode is useful for protecting traffic between different networks, when traffic must pass through an intermediate, un trusted network. Tunnel mode is primarily used for interoperability with gateways. An encrypted virtual tunnel is established between two communicating computers. In the tunnel mode, IPSec protect the entire IP datagram. It takes an IP datagram adds the IPSec header and trailer and encrypt the whole things. It then adds a new IP header to this encrypted datagram.

2. Transport Mode

Transport mode is the default mode for IPSec, and it is used for end-to-end cmmunications (for example, for communications between a client and a server). When transport mode is used, IPSec encrypts only the IP payload. In this mode it does not hide the actual source and destination address. They are visible in plain text. IPSec takes the transport layer payload adds IPsec header and trailer. Encrypt the whole thing send then adds IP header.

Authentication Header (AH)

- IPSec Authentication Header (AH) This protocol provides authentication services for IPsec. It allows the recipient of a message to verify that the supposed originator of a message was actually fact the one that sent it.
- It also allows the recipient to verify that intermediate devices route haven"t changed any of the data in the datagram. It also provides protection against so-called replay attacks, whereby a message is captured by an unauthorized user and resent.
- The IP Authentication Header (AH) is used to provide connectionless integrity and data origin authentication for IP datagrams and to provide protection against replays. AH provides authentication for as much of the IP header as possible, as well as for next level protocol data. However, some IP header fields may change in transit and the value of these fields, when the packet arrives at the receiver, may not be predictable by the sender. The values of such fields cannot be protected by AH. Thus, the protection provided to the IP header by AH is piecemeal.
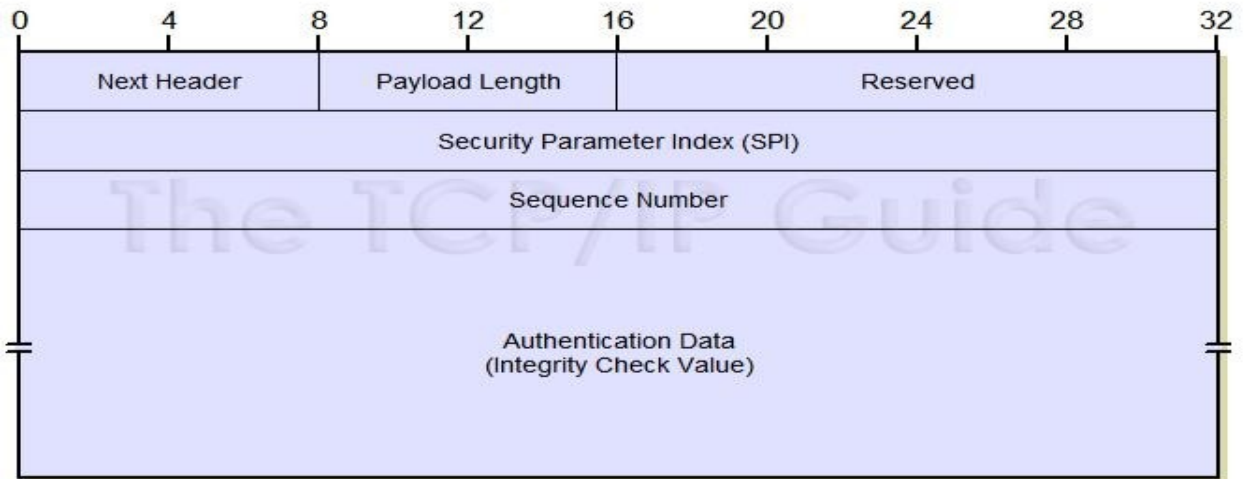
Fig 7.12Authentication Header format

- AH offers an anti-replay (partial sequence integrity) service at the discretion of the receiver, to help counter denial of service attacks. AH is an appropriate protocol to employ when confidentiality is not required (or is not permitted, e.g , due to government restrictions on use of encryption). AH also provides authentication for selected portions of the IP header, which may be necessary in some contexts. For example, if the integrity of an IPv4 option or IPv6 extension header must be protected en route between sender and receiver, AH can provide this service (except for the non-predictable but mutable parts of the IP header.)

AH tunnel mode

- As shown in the following illustration, AH tunnel mode encapsulates an IP packet with an AH and IP header and signs the entire packet for integrity and authentication.
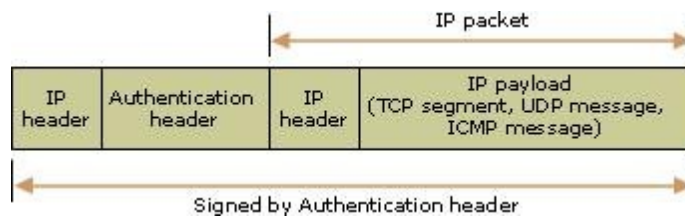
Authentication Header transport mode

- Authentication Header (AH) provides authentication, integrity, and anti-replay protection for the entire packet (both the IP header and the data payload carried in the packet). It does not provide confidentiality, which means that it does not encrypt the data. The data is readable, but protected from modification. AH uses keyed hash algorithms to sign the packet for integrity.
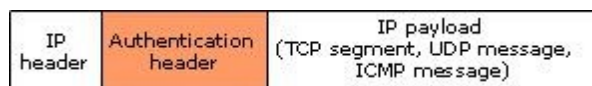
Packet signature with the AH header

- AH signs the entire packet for integrity, with the exception of some fields in the IP header which might change in transit (for example, the Time to Live and Type of Service fields). If another IPSec header is being used in addition to AH, the AH header is inserted before any other IPSec headers. The AH packet signature is shown in the following illustration.
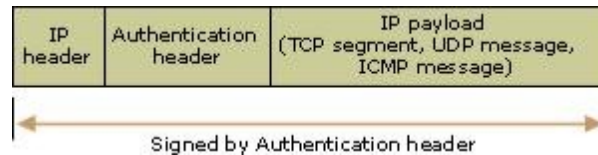
Encapsulating Security Payload (ESP)

- The Encapsulating Security Payload protocol provides encryption (confidentiality) and authentication (connectionless integrity and data origin authentication).
- ESP protects an IP packet but not additional headers that ESP adds.
- The Authentication Header protocol provides everything that ESP does –connectionless integrity and data origin authentication – but it does not encrypt to ensure confidentiality. AH protects an IP packet and also additional headers that AH adds.
- ESP and AH may be applied alone or together to provide the desired security services. The security they provide depends on the cryptographic algorithms they apply. Both are algorithm-independent, which lets new algorithms be added without affecting other parts of the implementation
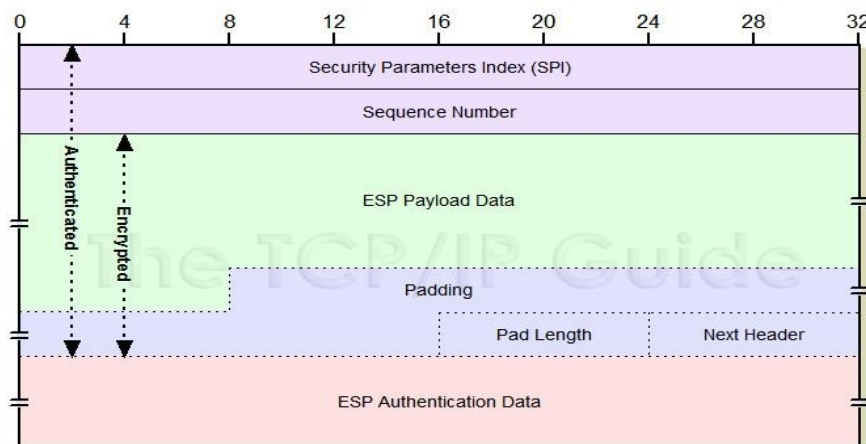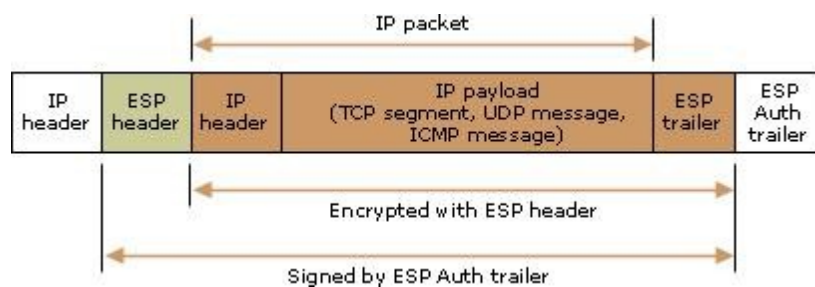

Fig 7.13 Encapsulating Security Payload (ESP) format

ESP tunnel mode

- ESP tunnel mode encapsulates an IP packet with both an ESP and IP header and an ESP authentication trailer.
- The signed portion of the packet indicates where the packet has been signed for integrity and authentication. The encrypted portion of the packet indicates what information is protected with confidentiality. Because a new header for tunneling is added to the packet,

everything that comes after the ESP header is signed (except for the ESP authentication trailer) because it is now encapsulated in the tunneled packet.

- The original header is placed after the ESP header. The entire packet is appended with an ESP trailer before encryption occurs. Everything that follows the ESP header, except for the ESP authentication trailer, is encrypted. This includes the original header which is now considered to be part of the data portion of the packet.

- The entire ESP payload is then encapsulated within the new tunnel header, which is not encrypted. The information in the new tunnel header is used only to route the packet from origin to tunnel endpoint. If the packet is being sent across a public network, it is routed to the IP address of the gateway for the receiving intranet. The gateway decrypts the packet, discards the ESP header, and uses the original IP header to route the packet to the intranet computer.

- ESP and AH can be combined when tunneling, providing both confidentiality for the tunneled IP packet and integrity and authentication for the entire packet.

Encapsulating Security Payload transport mode

- Encapsulating Security Payload (ESP) provides confidentiality (in addition to authentication, integrity, and anti-replay protection) for the IP payload.

- ESP in transport mode does not sign the entire packet. Only the IP payload (not the IP header) is protected. ESP can be used alone or in combination with AH.

- ESP is identified in the IP header with the IP protocol ID of 50., the ESP header is placed before the IP payload, and an ESP trailer and ESP authentication trailer is placed after the IP payload.
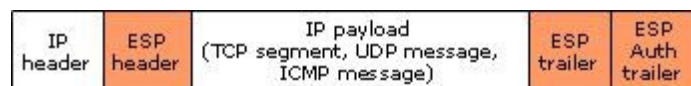

7.4 Virtual Private Network (VPN)

The commonly used initials "VPN" stand for the term Virtual Private Network. In its most basic definition a VPN is a network of computers which is kept private and secure despite being spread across unsecured public networks, such as the internet.

- They can be thought of in contrast to ring-fenced networks of computers behind a single firewall, situated in a single location, using dedicated on-site connections (i.e., LANs - Local Area Networks) or private networks of computers in disparate locations, connected using privately leased lines.

- VPNs therefore allow businesses and individuals to share sensitive information across computers, or other devices, in varying locations without the need to deploy distinct physical connections, and without compromising the security of those devices or their LANs.

- There are two broad classifications of VPN. The first, remote-access, describes a scenario in which an individual computing device establishes a connection with another or with an existing LAN.
- The second, site-to-site, involves two distinct LANs forming a connection across public networks to create a virtualised LAN.
- In practice VPNs can utilise a number of varying technologies and protocols to create secure connections on which data can be transferred. At their heart, though, lies the idea of creating a secure tunnel through a public network, within which all information can be passed; essentially a virtualised equivalent of a physical network connection or a leased line for example.

VPN Architecture

- A virtual private network transmits data through a secure broadband tunnel.
- ARCHITECHTURE VPNs essentially enhance the native Internet infrastructure with broadband connections and secured data platforms on the value-added networks so it can't be hacked into by unauthorized parties.
- So, they are high speed but secure. But because much of their physical connectivity is already provided by the public Internet infrastructure, they are much less expensive to build than private VANs and inherently more pervasive than leased lines. They essential combine the best characteristics of WANs, leased lines, and the Internet into an affordable, robust, and secure wide area infrastructure for data transfer .
  - A VPN consists of four components:
  - a VPN client
  - a network access server (NAS),
  - a tunnel terminating device (or VPN server),
  - and a VPN protocol.
- In a typical access VPN connection, a remote user (or VPN client) initiates a PPP connection with the ISP"s NAS via the public switched telephone network (PSTN).
- An NAS is a device that terminates dial-up calls over analog (basic telephone service) or digital (ISDN) circuits. The NAS is owned by the ISP, and is usually implemented in the ISP"s POP. After the user has been authenticated by the appropriate authentication method, the NAS directs the packet to the tunnel that connects both the NAS and the VPN server. The VPN server may reside in the ISP"s POP or at the corporate site, depending

- on the VPN model that is implemented. TheVPN server recovers the packet from the tunnel, unwraps it, and delivers it to the corporate network.
- There are four tunnelling protocols used to establish VPNs, and three are extensions of the Point-to-Point Protocol (PPP):
  - Point-to-Point Tunnelling Protocol (PPTP)
  - Layer 2 Forwarding (L2F)
  - Layer 2 Tunnelling Protocol (L2TP)
  - IP Security (IPSec) Protocol Suite