UNIT-1-Possible Attacks on Computers

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

INTRODUCTION:

- Computer data often travels from one computer to another, leaving the safety of its protected physical surroundings.
- Once the data is out of hand, people with bad intention could modify or forge your data, either for enjoyment or for their own benefit.
- Cryptography can reformat and transform our data, making it safer on its trip between computers.
- The technology is based on the secret codes, modern mathematics that protects our data in powerful ways.
- **Computer Security** - generic name for the collection of tools designed to protect data and to prevent hackers.
- **Network Security** - measures to protect data during their transmission.
- **Internet Security -** measures to protect data during their transmission over a collection of interconnected networks.

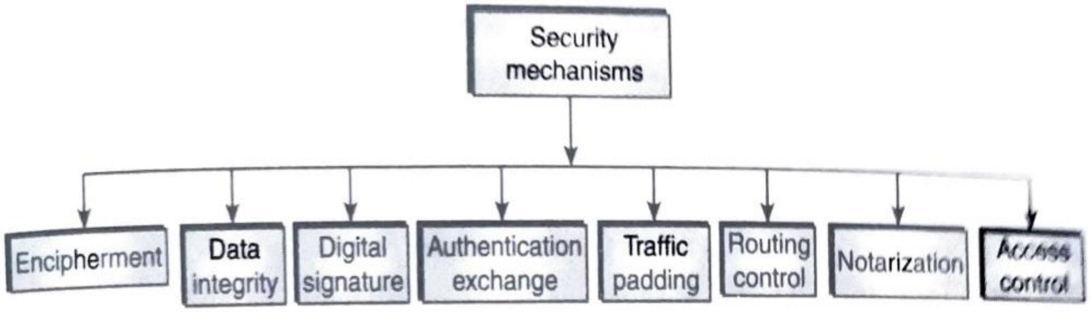
Why Do We Need Security?

In the ever-changing world of global data communications, inexpensive Internet connections and fast paced software development, security is becoming the biggest issue. Security is now a basic requirement because global computing is naturally insecure. As your data goes from point A to point B on the Internet, for example, it may pass through several other points along the way, giving other users the opportunity to intercept, and even alter it. It does nothing to protect your data center, other servers in your network, or a malicious user with physical access to your system.

Security Attacks, Services and Mechanisms:

To assess the security needs of an organization effectively, the manager responsible for security needs some systematic way of defining the requirements for security and characterization of approaches to satisfy those requirements. One approach is to consider three aspects of information security:

- **Security attack** – Any action that compromises the security of information owned by an organization.
- **Security mechanism** – A mechanism that is designed to detect, prevent or recover from a security attack.
- **Security service** – A service that enhances the security of the data processing systems and the information transfers of an organization. The services are intended to counter security attacks and they make use of one or more security mechanisms to provide the service.
- Security mechanisms have been defined by ITU-T (X 800). They used to implement security services. Some of the security mechanisms defined by ITU-T (X 800) are shown in the figure.



Encipherment: This refers to the transformation of the message or data with the help of mathematical algorithms. The main aim of this mechanism is to provide confidentiality. The two techniques that are used for encipherment are cryptography and steganography.
Data integrity: This refers to the method of ensuring the integrity of data. For this, the sender computes a check value by applying some process over the data being sent, and then appends this value to the data. On receiving the data, the receiver again computes the check value by applying the same process over the received data. If the newly computed check value is same as the received one, then it means that the integrity of data is preserved.
Digital signature: This refers to the method of electronic signing of data by the sender and electronic verification of the signature by the receiver. It provides information about the author, date and time of the signature, so that the receiver can prove the sender's identity.
Authentication exchange: This refers to the exchange of some information between two communicating parties to prove their identity to each other.
Traffic padding: This refers to the insertion of extra bits into the stream of data traffic to prevent traffic analysis attempts by attackers.
Routing control: This refers to the selection of a physically secured route for data transfer. It also allows changing of route if there is any possibility of eavesdropping on a certain route.
Notarization: This refers to the selection of a trusted third party for ensuring secure communication between two communicating parties.
Access control: It refers to the methods used to ensure that a user has the right to access the data or resource.

PRINCIPLES OF SECURITY/ SECURITY SERVICES:

The classification of security services are as follows:

Confidentiality:

- The principle of confidentiality specifies that only the sender and the intended recipient(s) should be able to access the contents of a message.
- Confidentiality gets compromised if an unauthorized person is able to access a message.
- Unauthorized party could be a person, a program or a computer.
- Example: Suppose a confidential email message sent by user A to user B, which is accessed by user C without the permission or knowledge of A and B. This type of attack is called interception.
- **Interception** causes loss of message confidentiality.

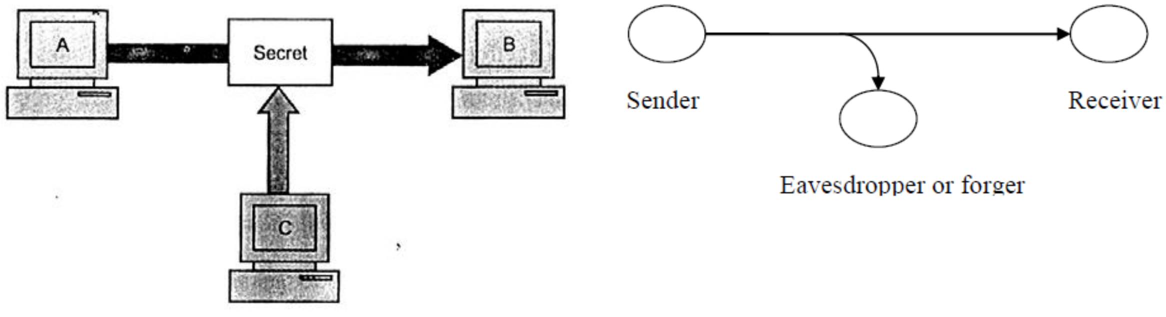


Fig.: Loss of confidentiality

Authentication

- Authentication mechanism helps to establish **proof of identities**.
- The authentication process ensures that the origin of a electronic message or document is correctly identified. This concept is shown in figure.
- **Fabrication** is possible in absence of proper authentication mechanisms.

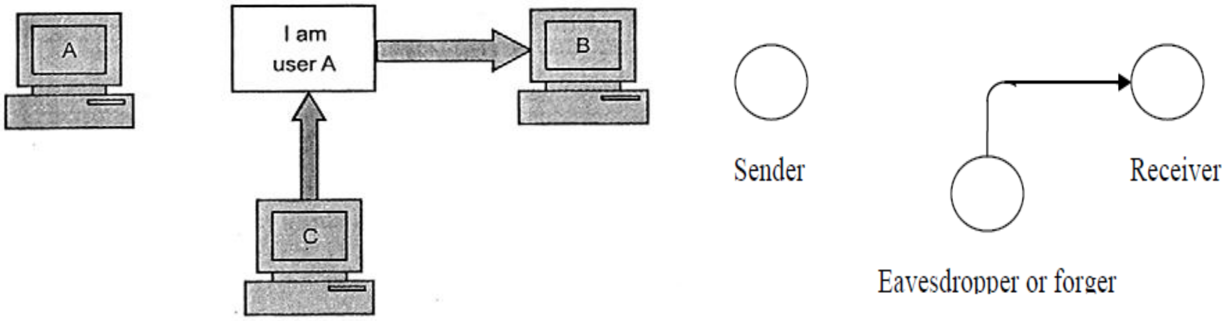


Fig.: Absence of authentication

Integrity

- When the contents of a message are changed after the sender sends it, but before it reaches the intended recipient, we say that the integrity of the message is lost. It is shown in figure.
- For example, consider that user A sends message to user B. User C tampers with a message originally sent by user A, which is actually meant for user B. User C change its contents and send the changed message to user B. User B has no way of knowing that the contents of the message changed after user A had sent it. User A also does not know about this change. This type of attack is called modification.
- **Modification** causes of loss of message integrity.

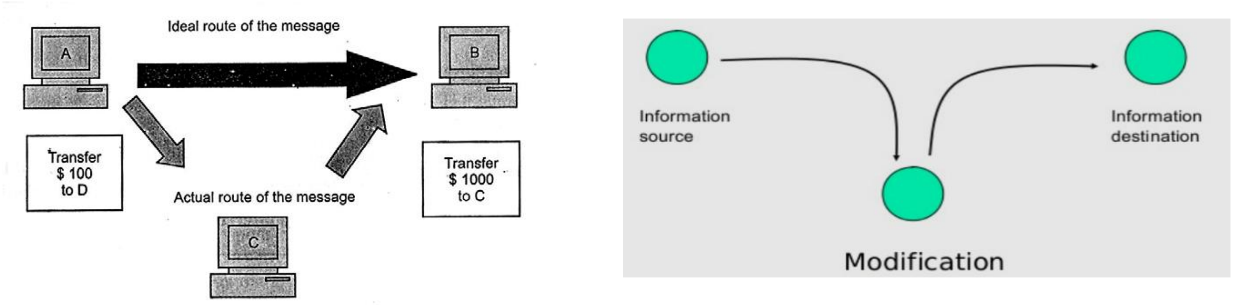


Fig.: Loss of integrity

Non repudiation:

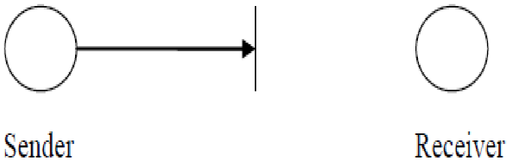
- Requires that neither the sender nor the receiver of a message be able to deny the transmission.

Access control:

- Access control determines and controls who can access what. It regulates which user has access to the resource, under what circumstances.

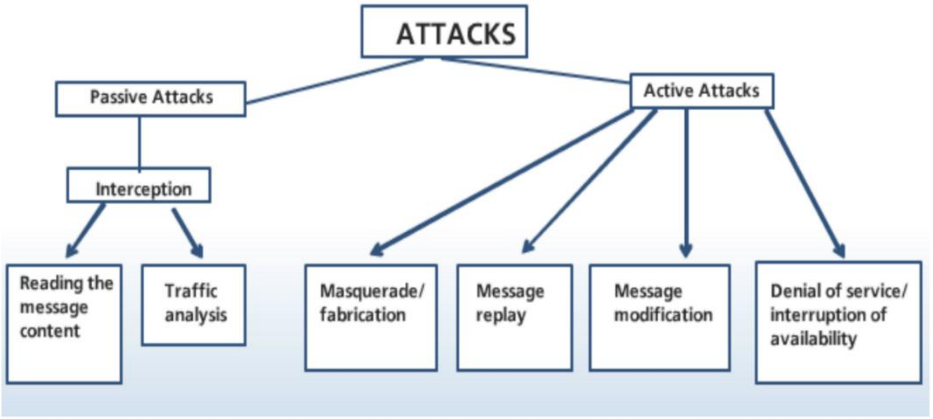
Availability:

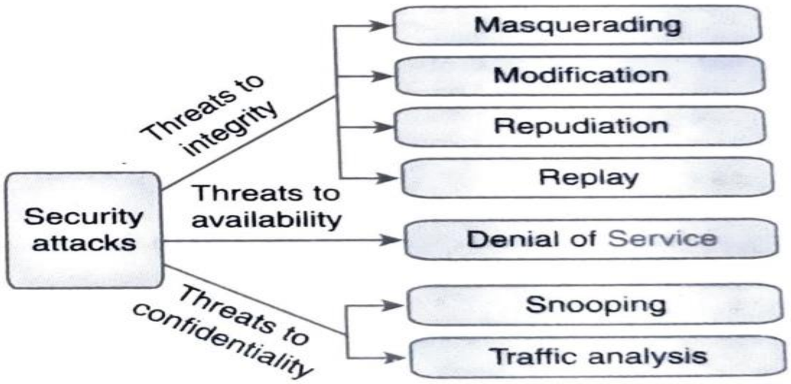
- The principle of availability is that resources should be available to authorized parties at all times.
- For example, due to the intentional actions of an unauthorized user C, an authorized user A may not be able to contact a server B. This would defeat the principle of availability. Such an attack is called interruption.
- **Interruption** causes loss of availability.



Types of Security attack:

- There are two types of attacks.
    - Active attacks
    - Passive attacks

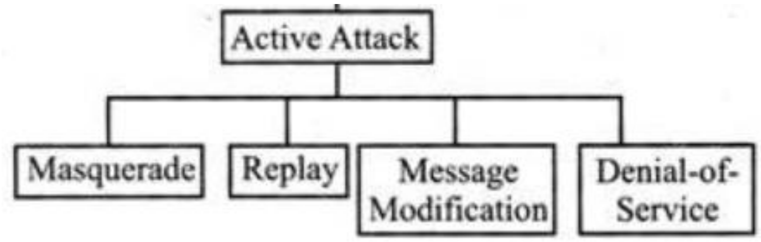




Active attacks

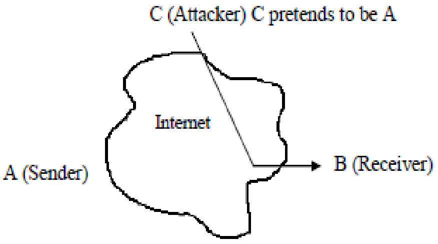
- An active attack is an attempt to alter system resources or affect their operation.
- I.e., these attacks involve in some modification to the original message in some manner or the creation of a false stream.
- These attacks can be classified in to four categories:

Sender
Receiver
Message
Modifying the message
Attacker
Active Attack



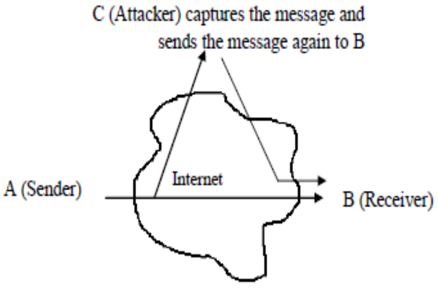
Masquerade:

- One entity pretends to be a different entity.
- It is generally done by using stolen IDs and passwords or through bypassing authentication mechanism.



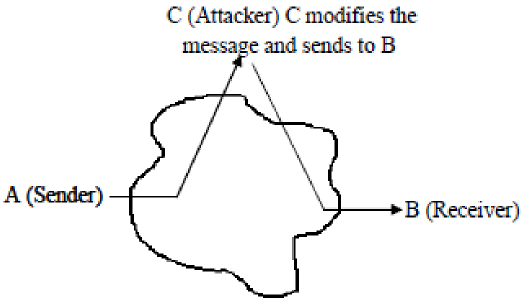
Replay:

- This attack involves capturing a copy of the message sent by the original sender and retransmitting it later to bring an unauthorized result.


C (Attacker) captures the message and sends the message again to B
Internet
A (Sender)
B (Receiver)

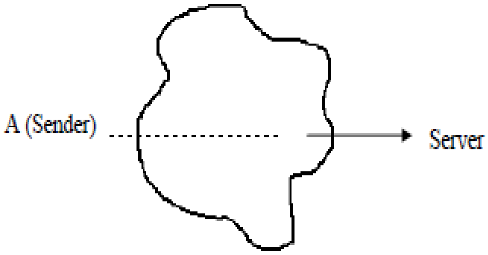
Modification of messages:

- Some portion of message is altered or the messages are delayed or recorded, to produce an unauthorized effect.
- For example, a message meaning "Allow John Smith to read confidential file *accounts*" is modified to mean "Allow Fred Brown to read confidential file *accounts*."



Denial of service:

- Prevents the normal use or management of communication facilities.
- Another form of service denial is the disruption of an entire network, either by disabling the network or overloading it with messages so as to degrade performance.



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

Passive Attacks

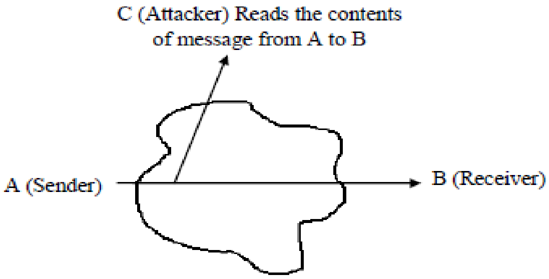
- A passive attack is an attempt to learn or make use of information from the system without affecting system resources.
- *Passive attacks* are those where the attacker indulges in eavesdropping or monitoring of data transmission.
- Passive attacks do not involve any modifications to the contents of an original message.

Message
Sender
Receiver
Observing the message
Attacker
Passive Attack

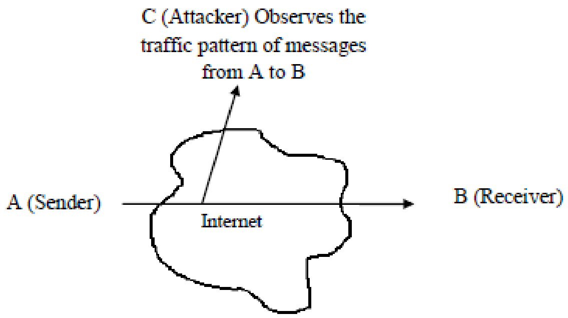
- There are two types of passive attacks.
  - Release of message contents and
  - Traffic analysis.

Release of message contents:

- It is easily understood by the given Figure.
- A telephone conversation, an electronic mail message, or a transferred file may contain sensitive or confidential information.
- We would like to prevent an opponent from getting the contents of these transmissions.



Traffic analysis:


C (Attacker) Observes the traffic pattern of messages from A to B
A (Sender)
Internet
B (Receiver)

- In this type of attack, an intruder observes the frequency and length of msg. being exchanged between communicating nodes.
- Attacker can then use this information for guessing the nature of communication that was taking place.

- Passive attacks are very difficult to detect because they do not involve any alteration of the data.
- Typically, the messages are sent and received in normal fashion. Neither the sender nor receiver is aware that a third party has read the messages or observed the traffic pattern.
- However, message encryption is a simple solution to prevent passive attacks. Thus, the emphasis in dealing with passive attacks is on prevention rather than detection.

UNIT: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

CRYPTOGRAPHY TECHNIQUES

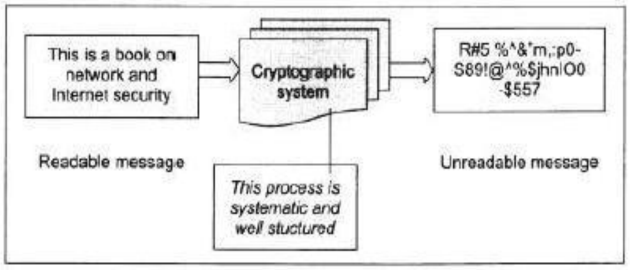
From the beginning any era, human being has two natural needs:

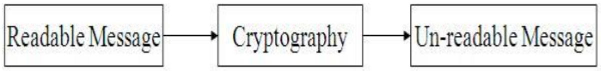
(a) To communicate and share information and
(b) To communicate selectively.

- These two needs gave rise to the art of coding the messages in such a way that only the intended people could have access to the information. Unauthorized people could not extract any information.
- The word "cryptography‟ is the combination of two Greek words, "Krypto" meaning hidden or secret and "graphene" meaning writing.

Cryptography: It is the art of achieving security by encoding messages to make them non-readable format.

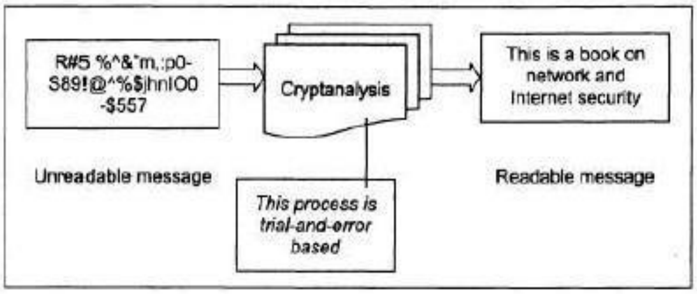
- ✓ The art or science encompassing the principles and methods of transforming an intelligible/understandable message into one that is unintelligible/non-understandable form, and then retransforming that message back to its original form.

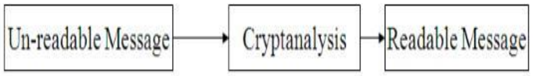

This is a book on network and Internet security
Cryptographic system
R#5 %^&*m,:p0-S89!@^%$jhnlO0 -$557
Readable message
Unreadable message
This process is systematic and well stuctured



Cryptanalysis: It is the technique of decoding messages from a non-readable format back to a readable format.

- ✓ It is done without knowing how they were initially converted from readable format to non-readable format.
- ✓ Also called code breaking.

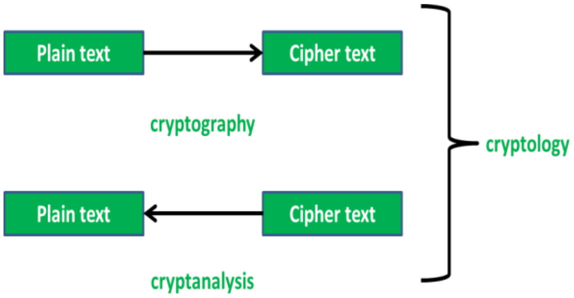

R#5 %^&*m,:p0-S89!@^%$jhnlO0 -$557
Cryptanalysis
This is a book on network and Internet security
Unreadable message
Readable message
This process is trial-and-error based



Cryptanalysis

Cryptology: is a combination of cryptography and cryptanalysis.

✓ Cryptography + Cryptanalysis = Cryptology.



Some basic terminologies used:

Plain Text:

- Also called as *clear text*/ normal text.
- Language that we normally use.
- Easily understood by everybody.
- Unencrypted message.

Example: Hi Amit.

Cipher Text:

- The coded message.
- Language that cannot be understood.
- To achieve security, plain text is transformed into cipher text.
- Encrypted message.

Example: Plain Text: Hi Amit.
Cipher Text: **Ki Dplw.**

Encryption/ Encipher/Encrypt:

- Converting plaintext to ciphertext.
- Encryption is the process of encoding a message or plain text so that ciphertext can be produced.
- Plaintext is converted into ciphertext by using encryption algorithm.
- Converting ciphertext into plaintext.
- Decryption is the reverse process, transforming an encrypted message back into its normal text/plaintext.
- This is done by using decryption algorithm.

Cipher:

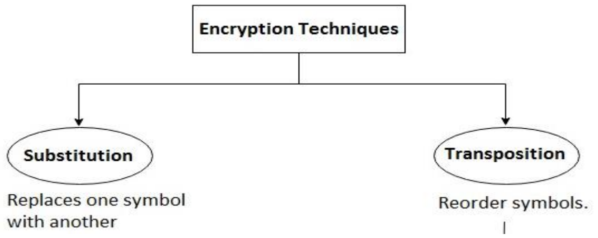
- Encryption and Decryption algorithms are together known as cipher.

Key:

- It is a number or set of numbers on which the cipher operates.

Encryption Technique/ transforming a Plaintext into Ciphertext:

- Clear text, or plain text, signifies a message that can be understood by the sender, the recipient, and also by anyone else who gets access to that message. When a plain-text message is codified using any suitable scheme, the resulting message is called ciphertext.



- There are two primary ways in which a plain-text message can be codified/ transform to obtain the corresponding ciphertext:
  - Substitution technique and
  - Transposition technique.

Substitution-cipher technique:

In the substitution-cipher technique, the each characters of a plain-text message are replaced by other characters, numbers or symbols.
There are several techniques. They are:

- Caesar Cipher
- Modified version of Caesar Cipher
- Monoalphabetic Cipher
- Homophonic Substitution Cipher
- Polygram Substitution Cipher
- Playfair Cipher
- Polyalphabetic Cipher
- Hill Cipher

Caesar Cipher

- Proposed by Julius Caesar.
- Mechanism to make a plaintext message into ciphertext message.
- It replacing each letter of the alphabet with the letter standing n places further down the alphabet.
- Example: Replace each A with D, B with E, etc.

ABCDEFGHIJKLMNOPQRSTUVWXYZ
DEFGHIJKLMNOPQRSTUVWXYZABC

PT: KIIT
CT: NLLW

How to break the Caesar cipher:

- All that is required to break the Caesar cipher is to do the reverse of the Caesar cipher process.
- I.e. replace each alphabet in a cipher-text message produced by Caesar cipher with the alphabet that is three places up the line.
- Thus, to work backwards, take a cipher text produced by Caesar cipher, and replace each **A** with **X**, **B** with **Y**, **C** with **Z**, **D** with **A**, **E** with **B** and so on.

Modified Version of Caesar Cipher

- The Caesar cipher is very simple and very easy to break. To make it complicated the modified version of Caesar cipher comes into play.
- Let us assume that the cipher-text alphabets corresponding to the original plain-text alphabets may not necessarily be three places down the order, but instead, can be *any* places down the order.

- As we know, the English language contains 26 alphabets. Thus, an alphabet A can be replaced by any *other* alphabet in the English alphabet set, (i.e. B through Z). Of course, it does not make sense to replace an alphabet by itself (i.e. replacing A with A).
- Thus, for each alphabet, we have 25 possibilities of replacement. Hence, to break a message in the modified version of Caesar cipher, our earlier algorithm would not work.

Mono-alphabetic Cipher

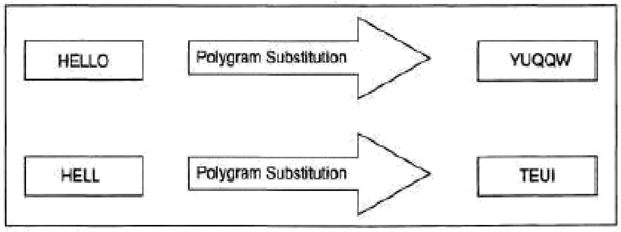
- A *monoalphabetic cipher* is a substitution cipher where a symbol in the plaintext has a one-to-one relationship with a symbol in the ciphertext.
- It means that a symbol in the plaintext is always replaced with the same symbol in the ciphertext, irrespective of its position in the plaintext.
- It uses random substitution.
- This means that in a given plain-text message, each A can be replaced by any other 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—we can replace each B with any other character!
- To put it mathematically, we can now have any permutation or combination of the 26 alphabets, which means (26 x 25 x 24 x 23 x ... 2) or 4 x 10^{26} possibilities! This is extremely hard to crack.

Homophonic Substitution Cipher

- This **substitution cipher** is very similar to mono-alphabetic cipher.
- However, the difference between the two techniques is in homophonic substitution cipher, one plain-text alphabet can map to more than one cipher-text alphabet.
- For instance, A can be replaced by <D, H, P, R>; B can be replaced by <E, I, Q, S> etc.

Polygram Substitution Cipher

- Polygram substitution cipher technique replaces one block of plain text with another block of cipher text—it does not work on a character-by-character basis.
- For instance, HELLO could be replaced by YUQQW, but HELL could be replaced by a totally different cipher text block TEUI, as shown in Fig.
- This is true in spite of the first four characters of the two blocks of text (HELL) being the same. This shows that in the polygram substitution cipher, the replacement of plain text happens block by block, rather than character by character.



Polygram substitution

Polyalphabetic Substitution Cipher

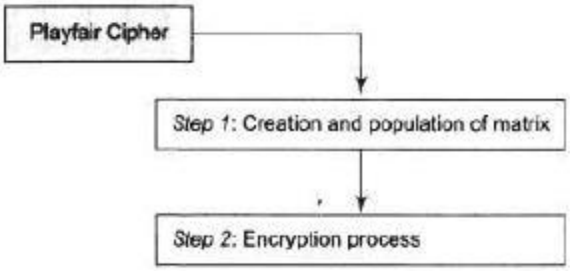
- Leon Battista invented the **polyalphabetic substitution cipher** in 1568.
- This cipher uses multiple one-character keys. Each of the keys encrypts one plain-text character. The first key encrypts the first plain-text character; the second key encrypts the second plain-text character, and so on.
- After all the keys are used, they are recycled. Thus, if we have 30 one-letter keys, every 30th character in the plain text would be replaced with the same key.

Playfair Cipher:

- The Playfair cipher, also called Playfair square, is a cryptographic technique. This scheme was invented by Charles Wheatstone in 1854.
- However, eventually the scheme came to be known by the name of Lord Playfair, who was Wheatstone‟s friend. Playfair made this scheme popular, and hence his name was used.
- The Playfair cipher was used by the British army in World War I and by the Australians in World War II.
- **The Playfair encryption scheme uses two main processes.**
  > Creation and population of matrix
  > Encryption process

Step 1: Creation and Population of Matrix

- The Playfair cipher makes use of a 5 x 5 matrix (table), which is used to store a *keyword* or *phrase* that becomes the *key* for encryption and decryption.
- The way this is entered into the 5 x 5 matrix is based on some simple rules:



Playfair Cipher

Step 1: Creation and population of matrix

Step 2: Encryption process

1. Enter the keyword in the matrix row-wise: left-to-right, and then top-to-bottom.

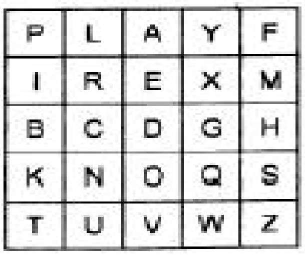
2. Drop duplicate letters.

3. Fill the remaining spaces in the matrix with the rest of the English alphabets (A-Z) that were not a part of our keyword. While doing so, combine I and J in the same cell of the table.

In other words, if I or J is a part of the keyword, disregard both I and J while filling the remaining slots.

EXAMPLE OF ENCRYPTION AND DECRYPTION IN PLAYFAIR:

For example, suppose that our **keyword=PLAYFAIR EXAMPLE**
Then, the 5 x 5 matrix containing our keyword will look as shown
Let us say, our **Plaintext= "MY NAME IS ATUL"**



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

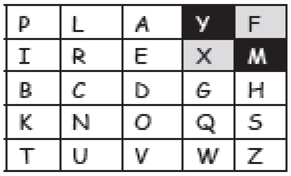
Encryption process – it consists of following steps:

1. Before initiating the encryption, break the plain text in pair of 2 letters.
   For ex. if our message is MY NAME IS ATUL, it becomes MY NA ME IS AT UL.
2. If both the alphabets are same or 1 letter is remaining, add X after the first alphabet.
3. After the initial process, take the pairs for encryption.
4. If the alphabets of the pair appear in same row of the matrix, then substitute them with their immediate right letter. If the alphabets of the plain text is itself the rightmost, then wrap it up with the left letter of the row it happens.
5. If the alphabets of the pair appear in same column of the matrix, then substitute them with their immediate below alphabets. If the letter of the plain text is itself below, then wrap it up with the top letter of the column it happens.
6. If the alphabets of the pair are not in same row or column then define a rectangle with the original pair and substitute them with other corners of the rectangle.

Example:

1) Message is: MY NAME IS ATUL It becomes MY NA ME IS AT UL.

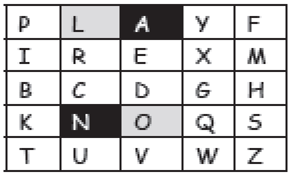
2) (step #5)



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

Cipher text - XF

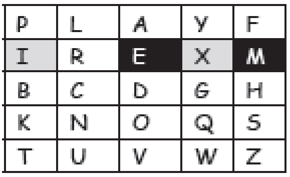
3) (step #5)



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

Cipher text - OL

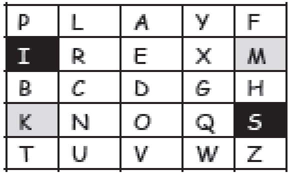
4) (step #3)



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

Cipher text - IX

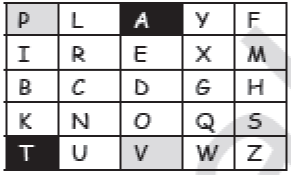
5) (step #5)



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

Cipher text - MK

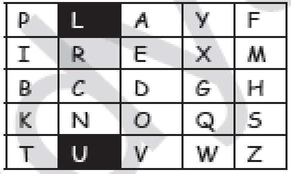
6) (step #5)



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

Cipher text - PV

7) (step #4)



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

Cipher text - LR

Plain text - MY NA ME IS AT UL
Cipher text - XF OL IX MK PV LR

Hill Cipher

- The **Hill cipher** works on multiple letters at the same time.
- Lester Hill invented this in 1929. The Hill cipher uses the matrix theory of mathematics.

Working:

- Treat each letter with a number like A=0, B=1, C=2…… Z=25.
- Let us say, our original message is “TAJ”
- As per the rule, T=19 A=0 J=9
- Convert it into matrix form as:

$$\begin{bmatrix}19\\0\\9\end{bmatrix}$$

Now ***multiply the plain text matrix with any number as keys***. The multiplying matrix should be of *n* x *n* where n is the number of rows of original matrix

$$\begin{bmatrix}6 & 24 & 1\\13 & 16 & 10\\20 & 17 & 15\end{bmatrix}\times\begin{bmatrix}19\\0\\9\end{bmatrix}=\begin{bmatrix}123\\337\\515\end{bmatrix}$$

Now compute ***mod 26*** on resultant matrix i.e. take the remainder after dividing by 26.

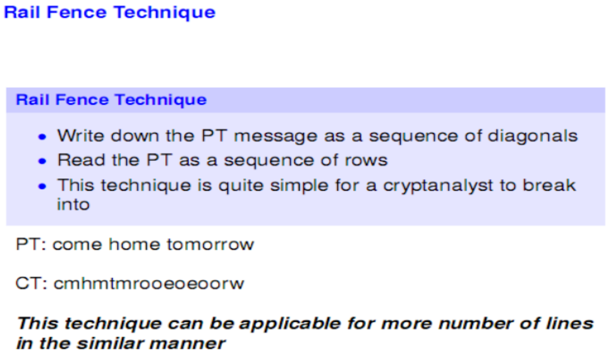
$$\begin{bmatrix}123\\337\\515\end{bmatrix}\bmod 26=\begin{bmatrix}19\\25\\21\end{bmatrix}$$

Now translating numbers into alphabets, we get:
19=T 25= Z 21=V
Therefore our cipher text is ***TZV***
To decrypt hill cipher, follow the steps:
1.) Take cipher text matrix and multiply it by inverse of original key matrix
2.) Again perform mod by 26.
Thus we get our original text.

Transposition techniques:

- In this cipher, there is no substitution of characters, it‟s position/ location of characters in plaintext is changed to place to form the ciphertext.
- For example, a symbol at the third position in the plaintext may be placed at the eighth position in the ciphertext, or a symbol at the fifth position in the plaintext may appear at the fifteenth position in the ciphertext.

Rail – fence technique:



Rail Fence Technique

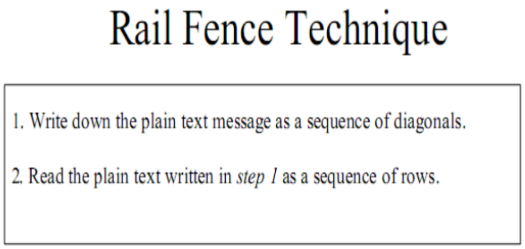
Rail Fence Technique

- Write down the PT message as a sequence of diagonals
- Read the PT as a sequence of rows
- This technique is quite simple for a cryptanalyst to break into

PT: come home tomorrow

CT: cmhmtmrooeoeoorw

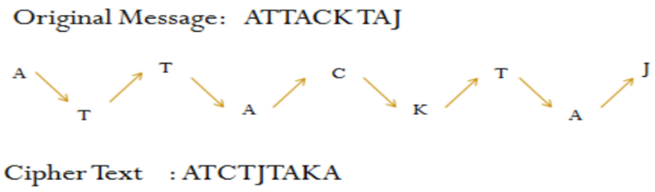
This technique can be applicable for more number of lines in the similar manner



Rail Fence Technique

1. Write down the plain text message as a sequence of diagonals.

2. Read the plain text written in *step 1* as a sequence of rows.

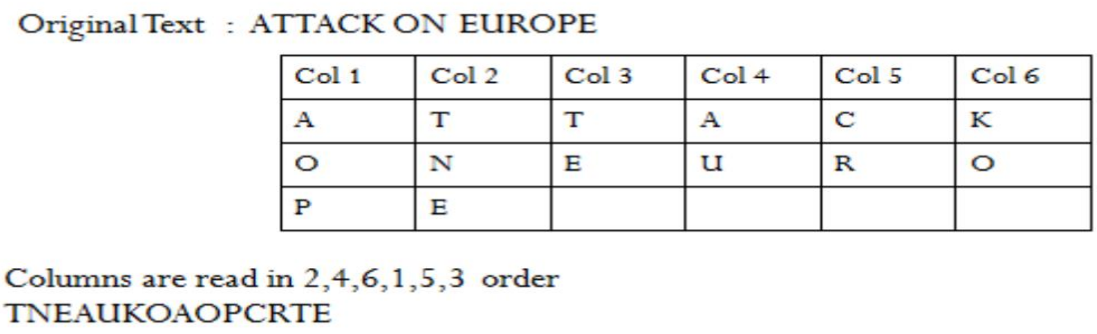


Original Message: ATTACK TAJ

Cipher Text : ATCTJTAKA

Simple Columnar Transposition Technique:

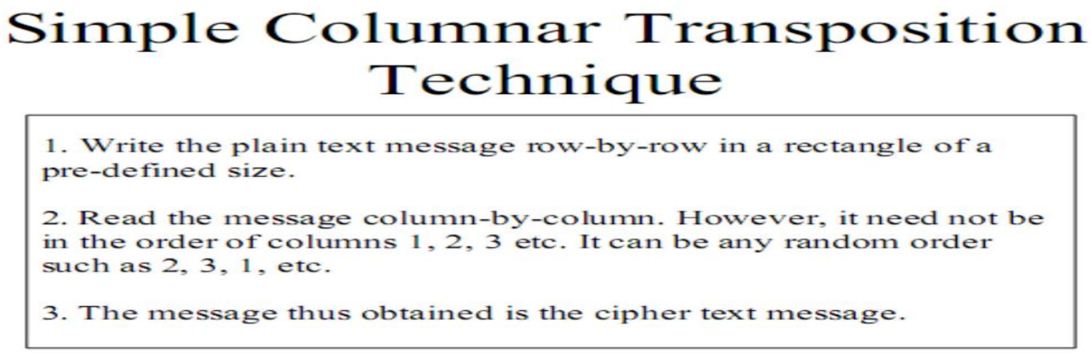
- Write the plain text message row by row in a rectangle of pre-defined size
- Read the message column by column but the sequence of columns can be any order.



Original Text : ATTACK ON EUROPE

Col 1	Col 2	Col 3	Col 4	Col 5	Col 6
A	T	T	A	C	K
O	N	E	U	R	O
P	E				

Columns are read in 2,4,6,1,5,3 order
TNEAUKOAOPCRTE



Simple Columnar Transposition Technique

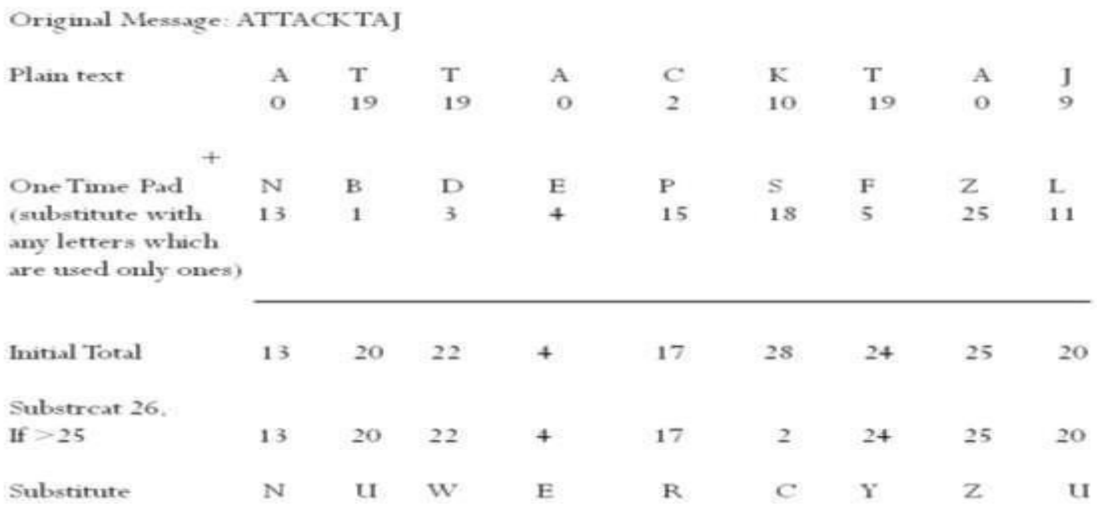
1. Write the plain text message row-by-row in a rectangle of a pre-defined size.

2. Read the message column-by-column. However, it need not be in the order of columns 1, 2, 3 etc. It can be any random order such as 2, 3, 1, etc.

3. The message thus obtained is the cipher text message.

Vernam Cipher (one time pad):

- It uses a one-time pad, which is discarded after a single use and therefore is suitable only for short messages.
- Treat each plain text alphabet with numbers as A=0, B=1, C=2……



Original Message: ATTACKTAJ

Plain text	A 0	T 19	T 19	A 0	C 2	K 10	T 19	A 0	J 9
+									
One Time Pad (substitute with any letters which are used only ones)	N 13	B 1	D 3	E 4	P 15	S 18	F 5	Z 25	L 11
Initial Total	13	20	22	4	17	28	24	25	20
Substreat 26, If >25	13	20	22	4	17	2	24	25	20
Substitute	N	U	W	E	R	C	Y	Z	U

UNIT-3- Symmetric and 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

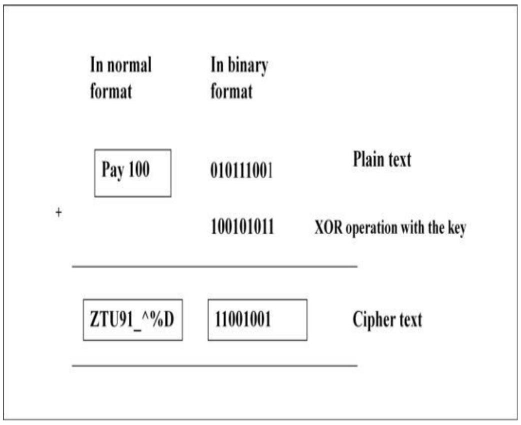
Algorithm Types and modes:

Algorithm types:

- It defines what size of plain text should be encrypted in each step of algorithm.
- It is of two types:
  - Stream Ciphers
  - Block Ciphers

Stream Ciphers

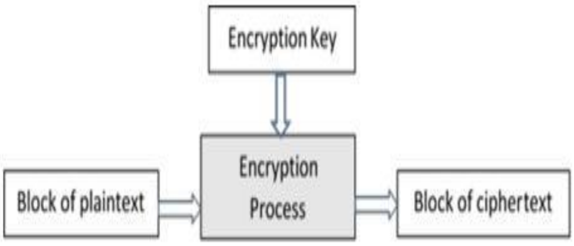
- Bit-by-bit encryption/decryption.
- In this scheme, the plaintext is processed one bit at a time i.e. one bit of plaintext is taken, and a series of operations is performed on it to generate one bit of cipher text.
- Technically, stream ciphers are block ciphers with a block size of one bit.
- Example: Suppose the original message (plain text) is Pay 100 in ASCII (i.e. text format).
- When we convert these ASCII characters to their binary values, let us assume that it translates to 01011100. Let us also Assume that we apply the XOR logic as the encryption algorithm.
- As a result of applying one bit of key for every respective bit of the original message, suppose the cipher text is generated as 11001001 in binary (ZTU91 A% in text).



Input 1	Input 2	Outputs
0	0	0
0	1	1
1	0	1
1	1	0

Block Cipher

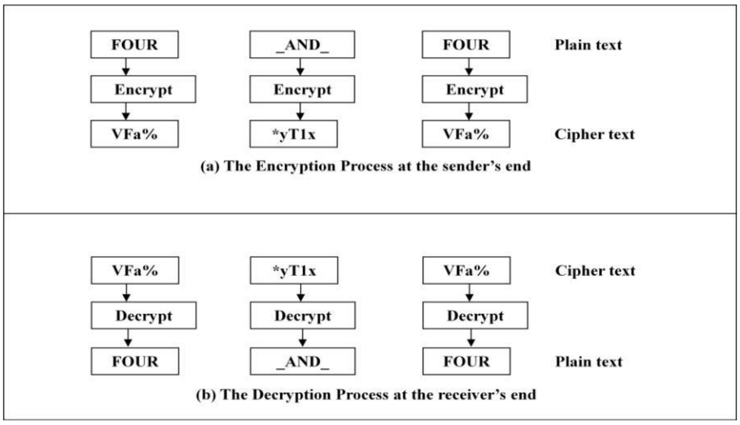
- Block-by-block encryption / decryption.
- In this scheme, the plain binary text is processed in blocks (groups) of bits at a time; i.e. a block of plaintext bits is selected, a series of operations is performed on this block to generate a block of cipher text bits.
- The number of bits in a block is fixed. For example, the schemes DES and AES have block sizes of 64 and 128, respectively.



The basic scheme of a block cipher is given as follows:

Block Cipher Example:

Suppose we have a plain text “FOUR_AND _FOUR” that needs to be encrypted. By using this technique FOUR could be encrypted first followed by _AND_ and FOUR.

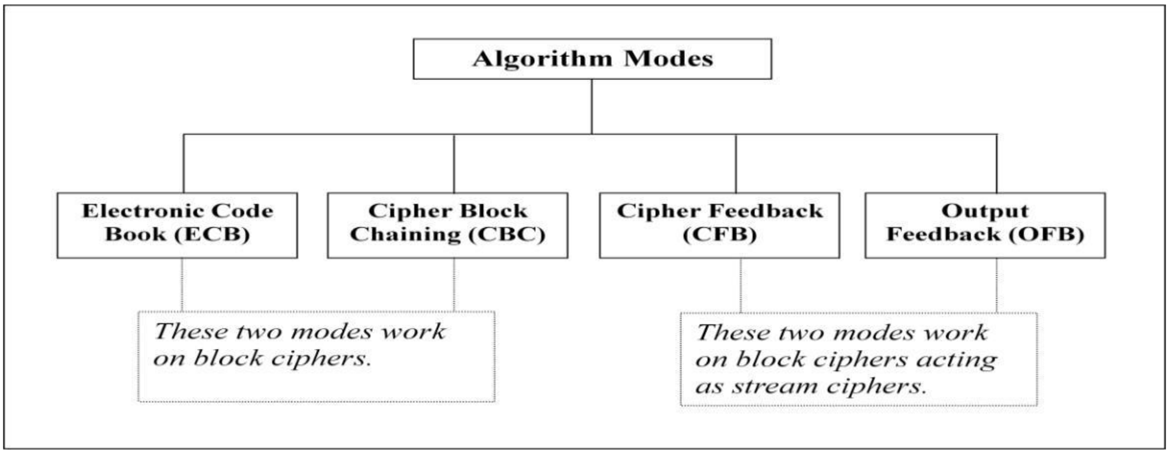


(a) The Encryption Process at the sender's end

(b) The Decryption Process at the receiver's end

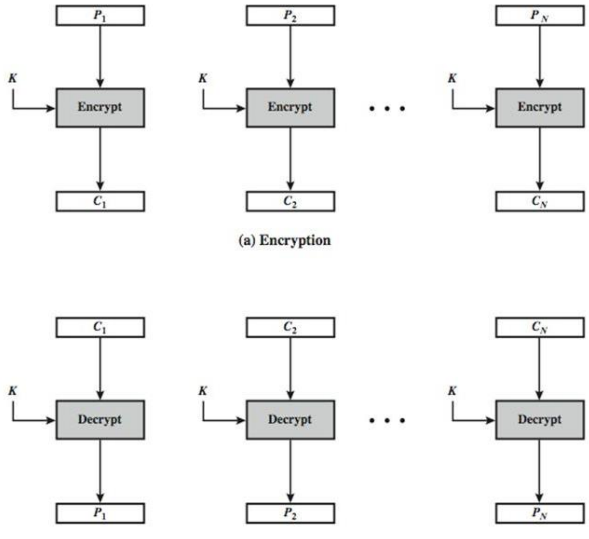
Algorithm Modes:

- It is a combination of series of basic algorithm steps on block cipher and some sort of feedback from the previous steps.
- It is divided into four modes:


Algorithm Modes
Electronic Code Book (ECB)
Cipher Block Chaining (CBC)
Cipher Feedback (CFB)
Output Feedback (OFB)
These two modes work on block ciphers.
These two modes work on block ciphers acting as stream ciphers.

Electronic Code book (ECB) Mode:

- ECB is a simplest and straightforward method of converting a block of plaintext into cipher text.
- Here, 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.
- This **encryption process** is shown figure.
- At the receiver's end, the incoming data is divided into 64-bit blocks.
- By using the same key as was used for encryption, each block is decrypted to produce the corresponding plain-text block.
- This **decryption process** is shown figure.



(a) Encryption

(b) Decryption

Cipher Block Chaining (CBC) Mode:

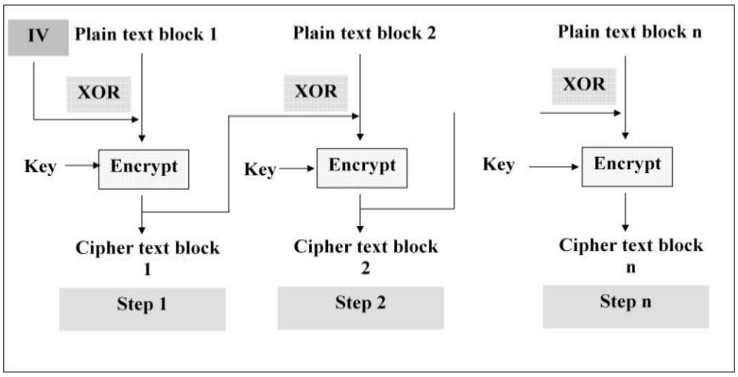
In CBC mode, a feedback mechanism is used. Chaining adds a feedback mechanism to a block cipher.

- In Cipher Block Chaining (CBC), the results of the encryption of the previous block are fed back into the encryption of the current block.
- That is, each block is used to modify the encryption of the next block.
- Thus, each block of cipher text is dependent on the corresponding current input plain-text block, as well as all the previous plain-text blocks.

Operation:

The steps are as follows:

- Load the n-bit Initialization Vector (IV). IV is a random generated block of text in a register.
- XOR the n-bit plain text block with data value in IV register.
- Encrypt the result of XOR operation with the key K. Result is it produce the cipher text block.
- Feed cipher text block into the IV register and continue the operation till all plaintext blocks are processed.

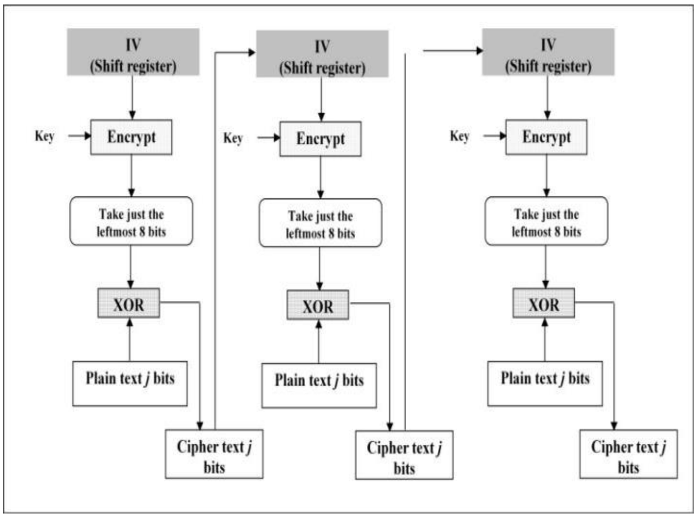


Cipher Feedback (CFB) Mode:

- Not all applications can work with blocks of data. Security is also required in applications that are character-oriented.
- For instance, an operator can be typing keystrokes at a terminal, which needs to be immediately transmitted across the communications link in a secure manner, i.e., by using encryption.
- In such situations, stream cipher must be used. The Cipher Feedback (CFB) mode is useful in such cases.
- In this mode, data is encrypted in units that are smaller (e.g., they could be of size 8 bits, i.e. the size of a character typed by an operator) than a defined block size (which is usually 64 bits).

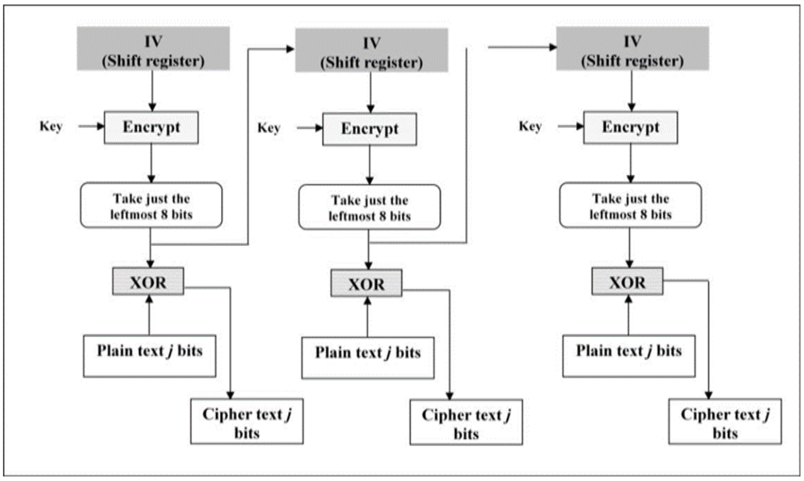
Steps of operation are:

- Assuming that we are dealing with j bits at a time (as we have seen usually, but not always, j = 8).
- we shall study CFB in a step-by-step fashion.
  - Step 1 Like CBC, a 64-bit Initialization Vector (IV) is used in the case of CFB mode. The IV is kept in a shift register. It is encrypted in the first step to produce a corresponding 64 bit cipher text.
  - Step 2 Now, the leftmost (i.e. the most significant) j bits of the encrypted IV are XORed with the first j bits of the plain text.
  - Step 3 Now, the bits of IV (i.e. the contents of the shift register containing IV) are shifted left by j positions. Thus, the rightmost j positions of the shift register now contain unpredictable data. These rightmost j positions are now filled with C.
  - Step 4 Now, steps 1 through 3 continue until all the plain-text units are encrypted.
- That is, the following steps are repeated:
  - IV is encrypted.
  - The leftmost j bits resulting from this encryption process are XORed with the next j bits of the plain text.
  - The resulting cipher-text portion (i.e., the next j bits of cipher text) is sent to the receiver.
  - The shift register containing the IV is left-shifted by j bits.
  - The j bits of the cipher text are inserted from right into the shift register containing the IV.



Output Feedback (OFB) Mode:

- The OFB mode is similar to CFB, but the only difference is that in CFB, the cipher text is fed into the next stage of encryption process.
- But in case of OFB the output of IV encryption process is fed into the next stage of encryption process.
- In this mode, if there are errors in individual bits, they remain errors in individual bits and do not corrupt the whole message.
- That is, bit errors do not get propagated.



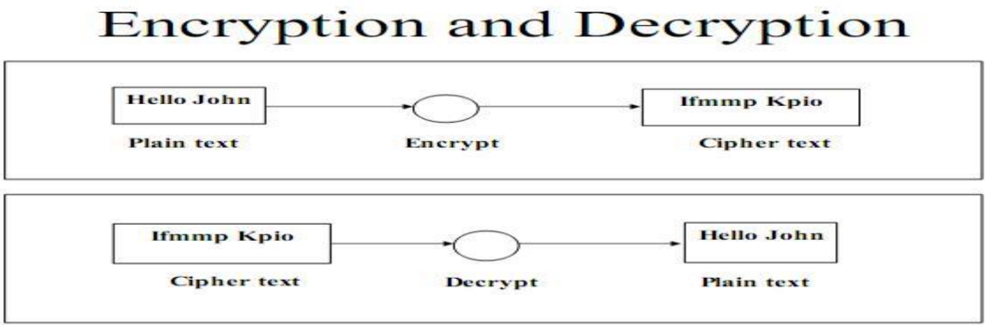
Encryption & Decryption:

Encryption or Encoding or Encode:

- The process of converting or transforming plain text or original text into cipher text is called as encoding.
- This new form of the message is totally different from the initial message.
- It occurs at the sender's side.
- The sender uses an encryption algorithm and a key to transform the original message into an encrypted message i.e., **cipher text**.
- Encryption is also called **enciphering or encipherment**.

Decryption or Decoding or Decode:

- The process of converting cipher text into plain text is called as decoding.
- It occurs at the receiver's end.
- The receiver uses decryption algorithms and a key to transform the cipher text back to original plaintext message.
- The decryption is also called **deciphering or decipherment**.
- Decryption is the reverse process of encryption.



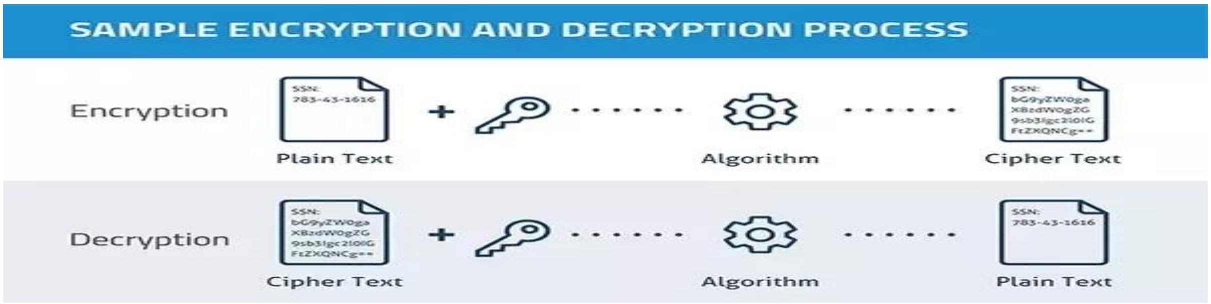
The important aspects of Encryption & Decryption process are:

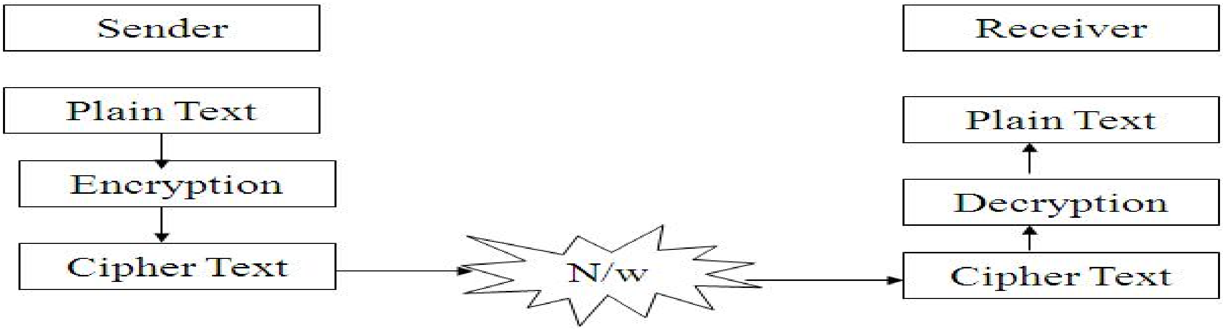
Algorithm:

- The technique/ method used to encrypt or decrypt. Algorithm is generally not kept secret.

Key:

- A key is a character or a group of characters used to encrypt or decrypt the plain text. A key is generally kept secret.


SAMPLE ENCRYPTION AND DECRYPTION PROCESS
Encryption
SSN: 783-43-1616
Plain Text
+
Algorithm
SSN: bG9yZWOga XBzdWOgZG 9sb3Igc2l0IG FtZXQNCg==
Cipher Text
Decryption
SSN: bG9yZWOga XBzdWOgZG 9sb3Igc2l0IG FtZXQNCg==
Cipher Text
+
Algorithm
SSN: 783-43-1616
Plain Text



Depending on what keys are used, there are two types of cryptography mechanisms/ types of cryptography:

Symmetric Key Cryptography:

- Symmetric key cryptography uses the same key for encryption and decryption.

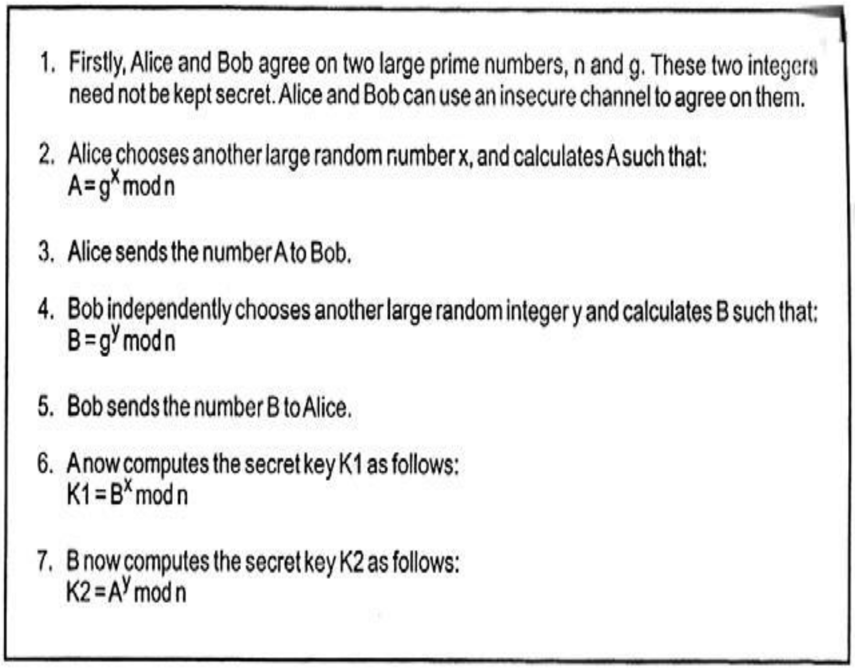
Asymmetric Key Cryptography:

- Asymmetric key cryptography uses one key for encryption, and another different key for decryption.

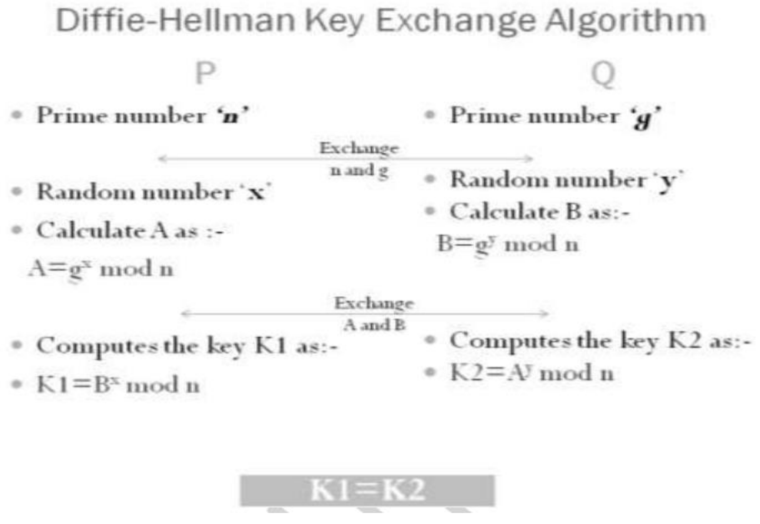
Diffie-Hellman Key Exchange/ Agreement Algorithm:

- Whitefield Diffie and Martin Hellman made a solution to the problem of key agreement, or key exchange. This solution is called the Diffie-Hellman key-exchange/agreement algorithm.
- 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.
- we must note that the 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 for actual encryption or decryption of messages.

Diffie-Hellman key exchange algorithm:

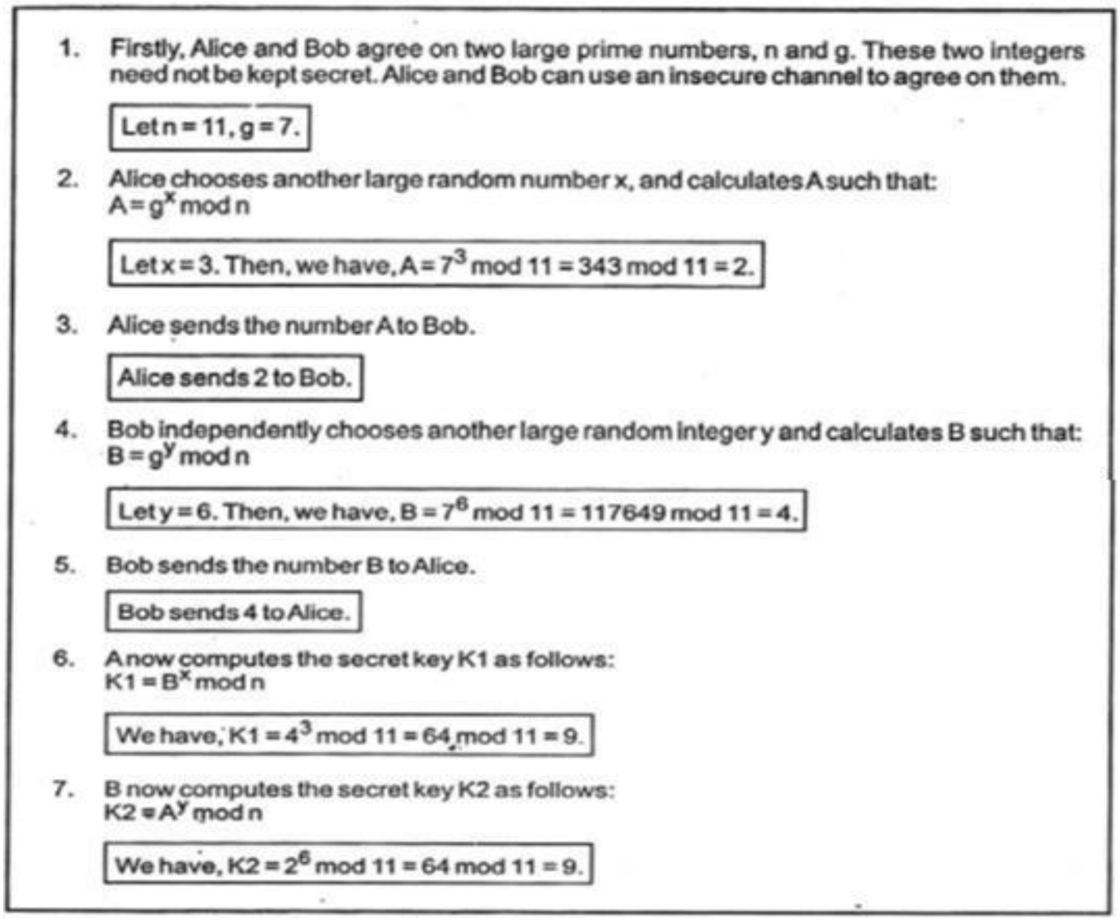


1. Firstly, Alice and Bob agree on two large prime numbers, n and g. These two integers need not be kept secret. Alice and Bob can use an insecure channel to agree on them.
2. Alice chooses another large random number x, and calculates A such that:
   $A = g^x \bmod n$
3. Alice sends the number A to 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:
   $K1 = B^x \bmod n$
7. B now computes the secret key K2 as follows:
   $K2 = A^y \bmod n$



Diffie-Hellman Key Exchange Algorithm

Example of the Algorithm: The process of key agreement in shown in below:



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, we have, $A = 7^3 \bmod 11 = 343 \bmod 11 = 2$.

3. Alice sends the number A to Bob.

   Alice sends 2 to Bob.

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, $B = 7^6 \bmod 11 = 117649 \bmod 11 = 4$.

5. Bob sends the number B to Alice.

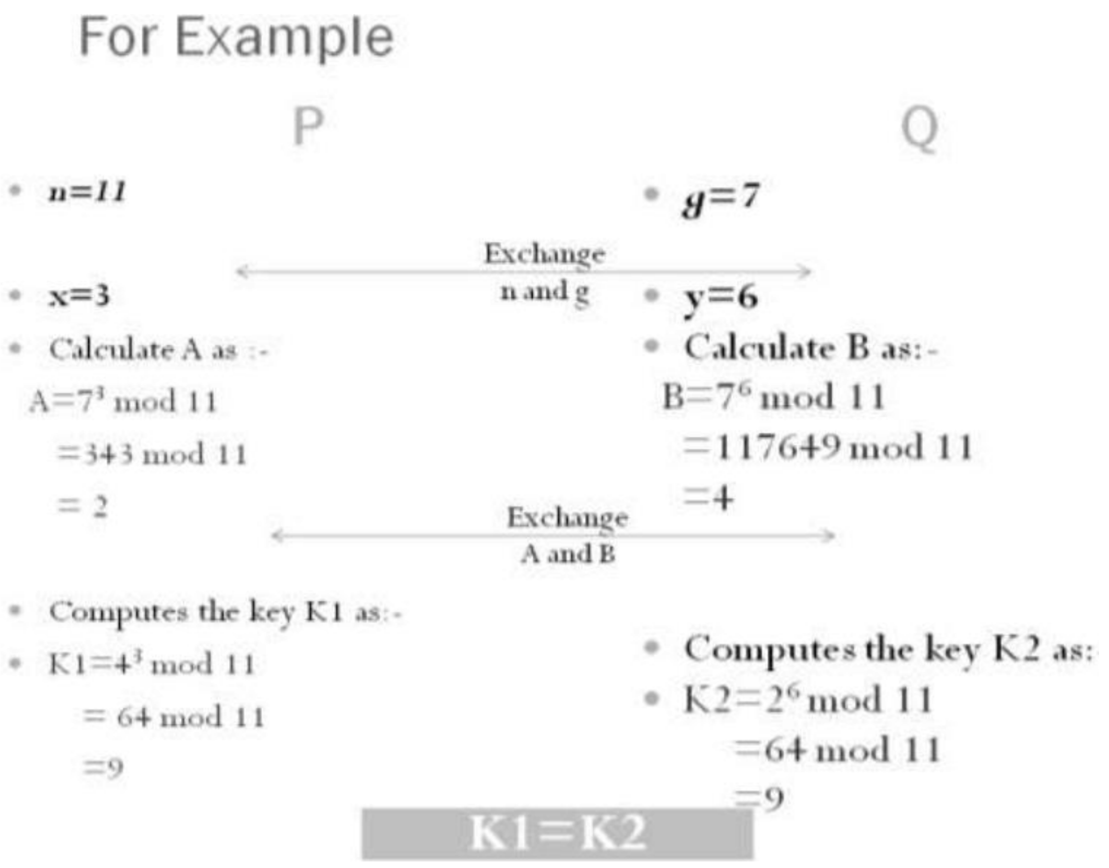
   Bob sends 4 to Alice.

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

   We have, $K1 = 4^3 \bmod 11 = 64 \bmod 11 = 9$.

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

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



For Example

P

Q

- $n=11$
- $g=7$

Exchange n and g

- x=3
- Calculate A as :-

  $A=7^3 \bmod 11$

  $=343 \bmod 11$

  $= 2$

- y=6
- Calculate B as:-

  $B=7^6 \bmod 11$

  $=117649 \bmod 11$

  $=4$

Exchange A and B

- Computes the key K1 as:-
- $K1=4^3 \bmod 11$

  $= 64 \bmod 11$

  $=9$

- Computes the key K2 as:
- $K2=2^6 \bmod 11$

  $=64 \bmod 11$

  $=9$

K1=K2

Symmetric-key/ Secret-key Encipherment/cryptography/cipher or Conventional encryption Model:

- The symmetric-key encipherment, sometimes also called secret-key encipherment or secret-key cryptography.
- It uses a single key known as shared key (or secret key) for both encryption and decryption of data.
- Thus, it is obvious that the key must be known to both the sender and the receiver.
- As shown in Figure, the sender uses the shared key and the encryption algorithm to transform the plaintext into cipher-text.
- The cipher-text is then sent to the receiver via a communication network.
- The receiver applies the same key and the decryption algorithm to decrypt the· cipher-text and to get the plaintext.
- Some examples of symmetric-key algorithms include Data Encryption Standard (DES), double DES, triple DES, and Advanced Encryption Standard (AES).

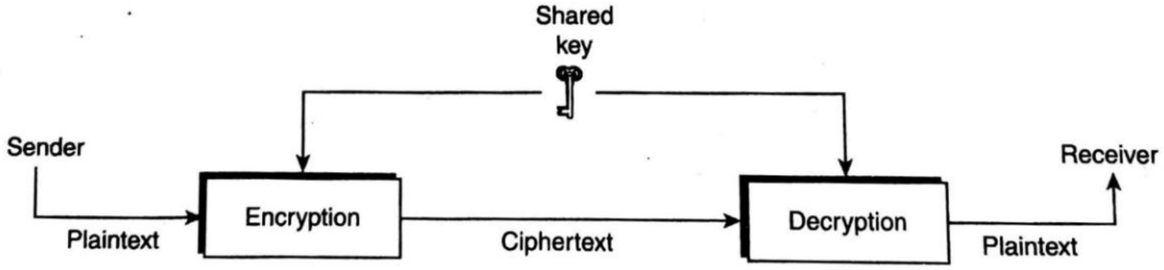


Figure 2.1 Message exchange using secret key

Secret-key encryption: it has different ingredients as shown in figure. They are:

- **Plaintext:** This is the readable or original message or data that is fed into the algorithm as input.
- **Cipher text:** This is the coded message produced as output (cipher texts or encrypted message) that is received by the receiver.
- **Encryption:** It is the process of encrypting the plaintext to produce the cipher-text. Plaintext is transformed into cipher-text using the encryption algorithm.
- **Decryption:** It is the reverse of the encryption process. In this process, the cipher-text is converted back to the plaintext using a decryption algorithm.
- **Ciphers:** The encryption and decryption algorithms are together known as ciphers.
- **Secret (shared) key:** This usually refers to a number or a set of numbers on which the cipher operates. Both encryption and decryption algorithms use the same key (shared between the sender and receiver) to encrypt or decrypt the messages, respectively.
- **Key:** A key is usually a number or a set of numbers on which the cipher operates. Encryption and decryption algorithms make use of a key to encrypt or decrypt messages, respectively.

- At the sender's end, the encryption algorithm and encryption key are required to convert the plaintext into cipher-text. At the receiver's end, a decryption algorithm and the decryption key to convert cipher-text back into the plaintext.
- The main problem in secret-key cryptography is that the sender and receiver have to agree on the secret key without anyone else finding it out.
- If the key is compromised, the security offered by secret-key cryptography is severely affected.
- Secret-key cryptography assumes that both parties who share a key rely upon each other and not to disclose the key and protect it against modification.
- If they are in separate physical locations, they must trust a medium such as the courier or a phone system to prevent the expose of the secret key. Anyone who hears or intercepts the key in transit can read, modify, and forge all messages using that key.

Asymmetric-key Enclpherment:

- The asymmetric-key encipherment also called public-key encipherment or public-key cryptography, was introduced by Diffie and Hellman in 1976 to overcome the problem found in symmetric key cryptography.
- It uses two different keys for encryption and decryption.
- These two keys are referred to as the public key (used for encryption) and the private key (used for decryption).
- Each authorized user has a pair of public and private keys. The public key of each user is known to everyone, whereas the private key is known to its owner only.

A **public-key encryption** scheme has six ingredients as shown in figure.

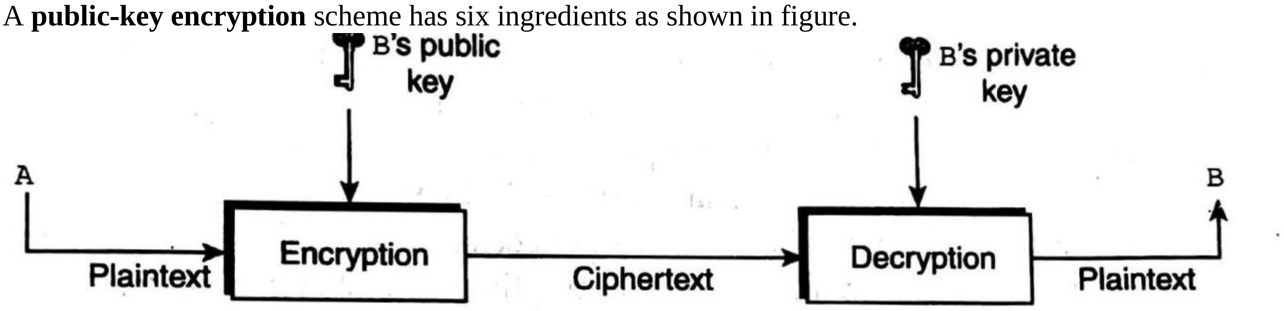


Figure 2.2 Message exchange using public key

- **Plaintext:** This is the readable message or data that is fed into the algorithm as input.
- **Encryption algorithm:** The encryption algorithm performs various transformations on the plaintext. This encrypts plain text using public key of receiver.
- **Public and Private keys:** This is a pair of keys used for encryption; the other is used for decryption. In figure, encryption is done using public key and decryption using private key.
- **Cipher-text:** This is the coded message produced as output. It depends on the plaintext and the key.
- **Decryption algorithm:** This algorithm accepts the cipher text and the matching key and produces the original plaintext. In figure, decryption algorithm uses private key.
- Now, suppose that a user ·A wants to transfer some information to· user B securely. The user A encrypts the data by using the public key of B and sends the encrypted message to B.
- On receiving the encrypted message, B decrypts it by using his/ her private key. Since decryption process requires a private key of user B, which is only known to B, the information is transferred securely. The above figure states the whole process.
- RSA is a well-known example of asymmetric-key algorithm.
- The main advantage of public-key cryptography is that the sender and the receiver need not have to share the secret key. All communication involves only public keys.
- Thus, the private key is never transmitted or shared. Anyone can send a confidential message using a public key, but the message can only be decrypted with a private key, which is kept by the intended recipient.

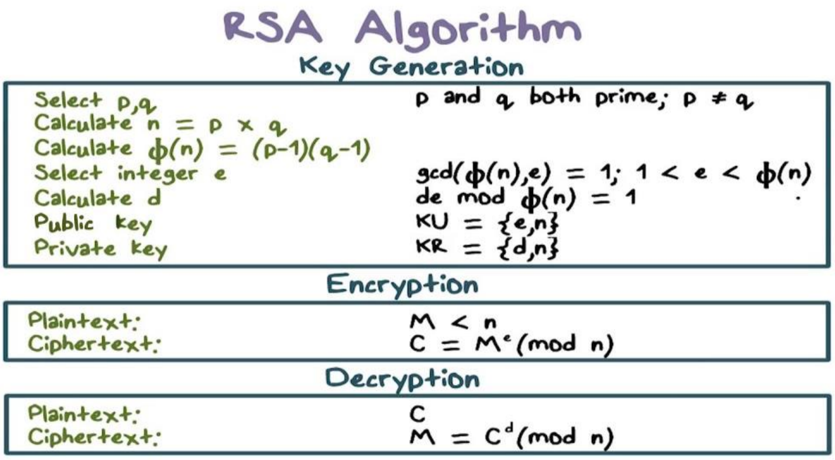
Differentiate between symmetric-key and asymmetric-key cryptography:

Symmetric-key	Asymmetric-key
1. It uses a single key for both encryption and decryption of data.	1. It uses .two different keys-public key for encryption and private key for decryption.
2. Both the communicating parties share the same algorithm and the key.	2. Both the communicating parties should have at least one of the matched pair of keys.
3.The processes of encryption and decryption are very fast.	3. The· encryption and decryption processes are slower as compared to symmetric-key cryptography.
4. Key distribution is a big problem.	4. Key distribution is not a problem.
5.The size of encrypted text is usually same or less than the original text.	5. The size of encrypted text is usually more than the size of the original text.
6.It can only be used for confidentiality, that is, only for encryption and decryption of data.	6. It can be used for confidentiality of data as well as for integrity and non-repudiation checks (i.e.for digital signatures).

THE RSA ALGORITHM:

This algorithm proposed by Ron Rivest, Adi Shamir, Len Adleman (RSA) in 1978 at MIT. It is based on asymmetric key cryptography.

1. Choose two large prime numbers *P* and Q.
2. Calculate *N = P x 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 x E) mod (P - 1) x (Q - 1) = 1
5. For encryption, calculate the cipher text *CT* from the plain text *PT* as follows:
   $CT = PT^E$ 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 follows:
   $PT = CT^D$ mod *N*



RSA Algorithm

Key Generation

Select p,q	p and q both prime; $p \neq q$
Calculate $n = p \times q$	
Calculate $\phi(n) = (p-1)(q-1)$	
Select integer e	$\gcd(\phi(n),e) = 1;\ 1 < e < \phi(n)$
Calculate d	$de \bmod \phi(n) = 1$
Public key	$KU = \{e,n\}$
Private key	$KR = \{d,n\}$

Encryption

Plaintext:	$M < n$
Ciphertext:	$C = M^e \pmod n$

Decryption

Plaintext:	C
Ciphertext:	$M = C^d \pmod n$

Examples of RSA

Let us take an example of this process to understand the concepts.

1. Choose two large prime numbers P and Q. Let P = 7, Q = 17.
2. Calculate N = P x Q.
   We have N = 7 x 17= 119.
3. Select the public key (i.e., the encryption key) E such that it is not a factor of (P - 1) X (Q - 1).
   Let us find (7 - 1) x (17 - 1) = 6 x 16 = 96.
   The factors of 96 are 2, 2, 2, 2, 2 and 3 (because 96 = 2 x 2 x 2 x 2 x 2 X 3).
   Thus, we have to choose E such that it is not the factors of E is 2 and 3.
   Let us choose E as 5 (it could have been any other number that does not its factors as 2 and 3).
4. Select the private key (i.e., the decryption key) D such that the following equation is true:
   (D x E) mod (P - 1) x (Q -1) = 1.
   Let us substitute the values of E, P and Q in the equation.
   We have: (D x 5) mod (7 - 1) x (17 - 1) = 1
   That is: (D x 5) mod (6) x (16) = 1
   That is: (D x 5) mod (96) = 1
   After some calculations, let us take D = 77. Then the following is true: (77 x 5) mod (96) = 385 mod 96 = 1.
5. For encryption, calculate the cipher text CT from the plain text PT as follows:
   $CT= PT^E$ mod N.

   Let us assume that we want to encrypt plaintext=10. Then we have:
   $CT = 10^5$ mod 119 = 100000 mod 119 = 40.

6. Send CT as the cipher-text to the receiver. Send 40 as the cipher text to the receiver.
7. For decryption, calculate the plaintext PT from the cipher-text CT as follows:
   PT = CT^D mod N.
   That is: PT = 40^{77} mod 119 = 10.
   This was the original plaintext of step 5.

Digital signature:

- It is an authentication mechanism that allows the sender to attach an electronic code with the message. This electronic code acts as the signature of the sender and hence, is named digital signature.
- It is done to ensure its authenticity and integrity.
- Digital signature uses the public-key cryptography technique. The sender uses his or her private key and a signing algorithm to create a digital signature and the signed document can be made public. The receiver, uses the public key of the sender and a verifying algorithm to verify the digital signature.
- A normal message authentication scheme protects the two communicating parties against attacks from a third party (intruder). However, a secure digital signature scheme protects the two parties against each other also.
- Suppose A wants to send a signed message (message with A's digital signature) to B through a network. For this, A encrypts the message using his or her private key, which results in a signed message. The signed message is then sent through the network to B.
- Now, B attempts to decrypt the received message using A's public key in order to verify that the received message has really come from A.
- If the message gets decrypted, B can believe that the message is from A. However, if the message or the digital signature has been modified during transmission, it cannot be decrypted using A's public key. From this, B can conclude that either the message transmission has tampered with, or that the message has not been generated by A.

Message integrity:

- Digital signatures also provide message integrity.
- If a message has a digital signature, then any change in the message after the signature is attached will invalidate the signature.
- That is, it is not possible to get the same signature if the message is changed. Moreover, there is no efficient way to modify a message and its signature such that a new message with a valid signature is produced.

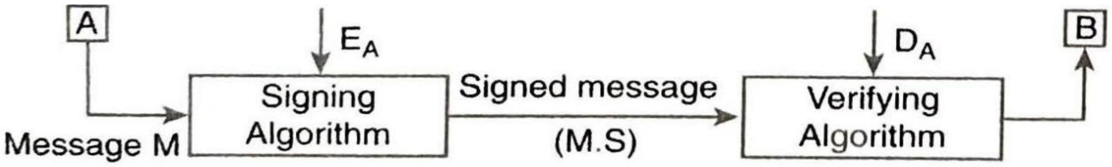
Non-repudiation:

- Digital signatures also ensure non-repudiation.
- For example, if A has sent a signed message to B, then in future A cannot deny about the sending of the message. B can keep a copy of the message along with A's signature.
- In case A denies, B can use A’s public key to generate the original message. If the newly created message is the same as that initially sent by A, it is proved that the message has been sent by A only.
- In the same way, B can never create a forged message bearing A's digital signature, because only A can create his or her digital signatures with the help of that private key.

Message confidentiality:

- Digital signatures do not provide message confidentiality, because anyone knowing the sender's public key can decrypt the message.

  Note: (this is additional) To achieve message confidentiality, we need to encrypt the message along with the signature using either the secret-key encryption or public-key encryption scheme. For example, if we use the public-key encryption scheme, then at A's end, first the message is encrypted using A's private key and then a second encryption is performed using the B's public key. Similarly, at B's end, first the message is decrypted using B's private key and then a second decryption is performed using A's public key. With this mechanism, only B can decrypt the encrypted message received from A because only B knows his or her private key.



Digital signature process:

The digital signature process is shown in Figure. Suppose user A wants to send a signed message to B through a network. To achieve this communication, these steps are followed:

- A uses his private key (EA), applied to a signing algorithm, to sign the message (M).
- The message (M) along with A's digital signature (S) is sent to B.
- On receiving the message (M) and the signature (S), B uses A's public key (DA), applied to the verifying algorithm, to verify the authenticity of the message. If the message is authentic, B accepts the message, otherwise it is rejected.

UNIT-4- Digital Certificate & Public Key Infrastructure

4.1 Digital Certificates	**4.3 The PKIX Model**
4.2 Private Key Management	**4.4 Public Key Cryptography Standards (PKCS)**

Digital Certificate

- To solve the man-in-the-middle attack, Digital Certificates were introduced.
- A digital certificate is simply a small computer file. For example, my digital certificate would actually be a computer file with a file name such as name .cer.
- The digital certificate is actually quite similar to a passport. As we know every passport has a unique passport number, similarly every digital certificate has a unique serial number. Also gives information of the issuer’s name, serial number, public key, validity period, etc.
- Digital Certificate is issued by **a trusted agency called as CA (Certification Authority).**
- Another third party called as RA (Registration Authority) acts as a intermediate entity between CA and end user.
- Satisfies the principle of Authentication, non-repudiation.

Who can be a CA?

- CA has to be someone, who everybody trusts. Consequently, the governments in various countries decide who can and who cannot be a CA.
- Usually, a CA is a reputed organization, such as a post office, financial institution, software company, etc. Two of the world’s most famous CAs are VeriSign and Entrust. Safescrypt Limited is the first Indian CA.
- Thus, a CA has the authority to issue digital certificates to individuals and organizations, who want to use those certificates in asymmetric-key cryptographic applications.

Technical Details of a Digital Certificate

A standard called **X.509** defines the structure of a digital certificate. The International Telecommunication Union (ITU) designs this standard. At that time, it was a part of another standard called **X.500**. The current version of the standard is Version 3, called **X.509V3.**

Contents of Digital Certificate:

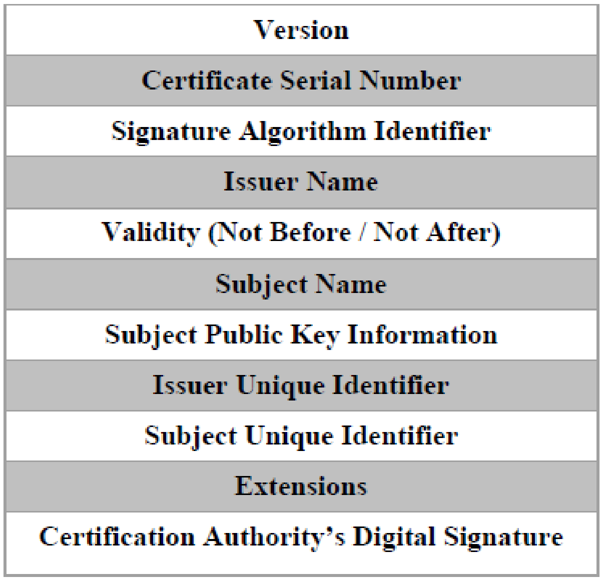
Version: Version of X.509 protocol. Version can be 1,2 or 3

Certificate Serial No.: Contains unique integer which is generated by CA

Signature Algorithm Identifier: Identifies the algorithm used by CA to sign the certificate.

Issuer Name: Identifies the Distinguished Name that created & signed the certificate

Validity: (not before/not after) Contains two date-time values. This value generally specifies the date & time up to seconds or milliseconds.



Version
Certificate Serial Number
Signature Algorithm Identifier
Issuer Name
Validity (Not Before / Not After)
Subject Name
Subject Public Key Information
Issuer Unique Identifier
Subject Unique Identifier
Extensions
Certification Authority’s Digital Signature

Subject name: Distinguished Name of the end user (user or organization)
Subject Public key info.: This field can never be blank. Contains public key & algorithm related.
Issuer Unique Identifier: Helps identify a CA uniquely if two or more CAs have used the same *IssuerName* over time.
Subject Unique Identifier: Helps identify a subject uniquely if two or more subjects have used the same *SubjectName* over time.

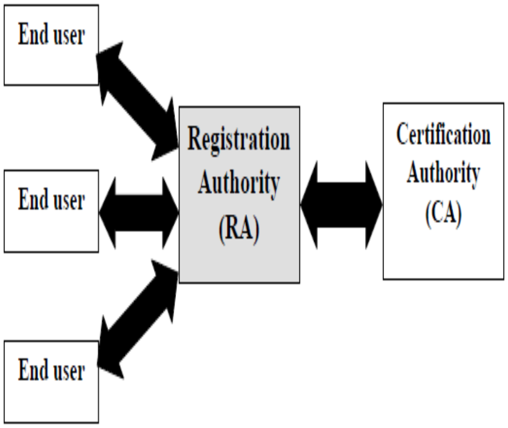
Digital-Certificate Creation:

1. ***Parties Involved***

- end user (may be a single user or organization),
- issuer (CA),
- third party is also (optionally) called a Registration Authority (RA), involved in the certificate creation and management.

The RA commonly provides the following services

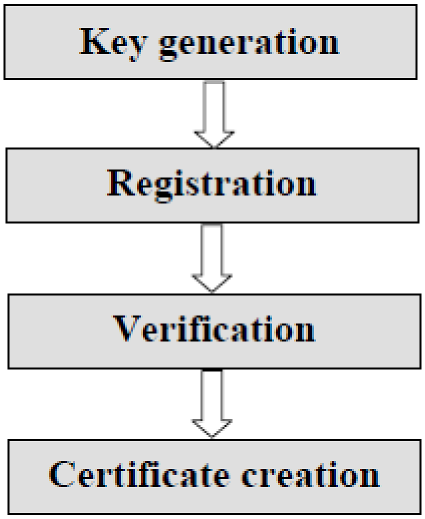
- Accepting and verifying registration information about new users.
- Generating keys on behalf of the end users.
- Accepting and authorizing requests for key backups and recovery.
- Accepting and authorizing the requests for certificate revocation.
- RA is mainly set up for facilitating the interaction between the end users and the CA
- The RA cannot issue digital certificates.
- The CA must handle this. Additionally, after a certificate is issued, the CA is responsible for all the certificate management aspects, such as tracking its status, issuing revocation notices if the certificate needs to be invalidated for some reason, etc.



2. ***Certificate Creation Steps***

Step 1: 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:
- Firstly, the subject can create a private key and public key pair using some software:
  - The subject must keep the private key which is generated, keep it secret. The subject then sends the public key along with other information to the RA.



- Secondly, the RA can generate a key pair on-behalf the subject.
    - This can happen in cases where either the user is not aware of the technicalities involved in the generation of a key pair.
    - The RA sends the private key which is generated, to the subject. The RA keeps the public key.

Step 2: Registration:

- This step is required only if the user generates the key pair in the first step. If the RA generates the key pair on the user's 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 associated registration information (e.g. subject name, as it is desired to appear in the digital certificate) and all the required evidence about himself/herself to the RA.
- For this, the software provides a wizard in which the user enters all the data then submits it. This data then travels over the network/Internet to the RA. This format for the certificate requests has been is called **Certificate Signing Request (CSR).** This is one of the **Public Key Cryptography Standards (PKCS),**
- Note that the user must not send the private key to the RA—the user must keep it securely.

Step 3: Verification:

After the registration process is complete, the RA has to verify the user's credentials. This verification is in two respects, as follows.

1. Firstly, the RA needs to verify the user's credentials which are provided by the user.
    - If the user were actually an **organization** then the RA would perhaps like to check the business records, historical documents and credibility proofs.
    - If it is an **individual** user then simpler checks are in call, such as verifying the postal address, email id, phone number, passport or driving-license details can be sufficient.
2. Secondly, check is to ensure that the user who is requesting for the certificate, whether he/she possesses the private key or not corresponding to the public key that is sent 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. Otherwise, this can create legal problems. This check is called the **Proof Of Possession (POP)** of the private key.

How can the RA perform this check? There are many approaches to this, the chief ones being as follows.

- The RA can demand that the user must digitally sign his/her Certificate Signing Request (CSR) using his/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 the user indeed possesses the private key.
- Alternatively, 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 his/her private key, the RA can assume that the user possesses the right private key.
- Thirdly, the RA can actually generate a dummy certificate for the user, encrypt it using the user's public key and send it to the user. The user can decrypt it only if he/she can decrypt the encrypted certificate, and obtain the plain-text certificate.

Step 4: Certificate Creation:

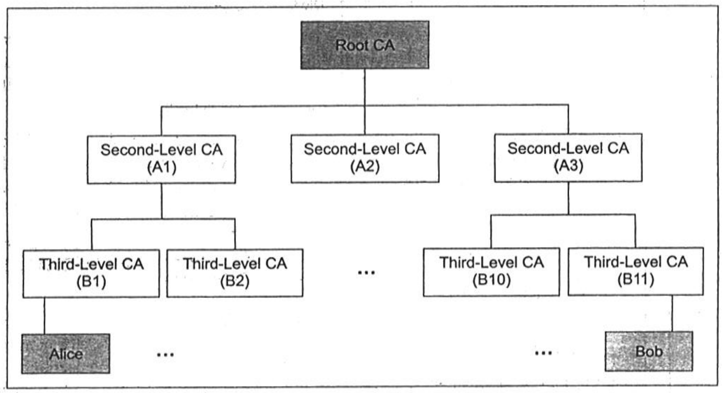
- Assuming that all the steps so far have been successfully done, and then RA passes on all the details of the user to the CA.
- The CA does its own verification (if required) and creates a digital certificate for the user.
- The creation of certificate as per the X.509 standard.
- The CA sends the certificate 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.

Certificate Hierarchies and Self-signed Digital Certificates:

Certificate hierarchy relieves the root CA from having to manage all the possible digital certificates.
As a substitute, the root CA can hand over this job to the second-level CAs. This hand over can happen region-wise. E.g. one second level CA could be responsible for the Western region, another for the Eastern region, a third one for the Northern region, and a fourth one for the Southern region, etc.). Each of these second-level CAs could appoint third-level CAs state-wise within that region. Each third-level CA could hand over its responsibilities to a fourth-level CA city-wise, and so on.
The root CA signs its own certificate. This certificate of the root CA is called **self-signed certificate.**

Cross-Certification

- It is quite possible that user A and user B live in different countries.
- This would mean that their root CAs may be different. Because generally each country appoints its own root CA. In fact, one country can have multiple root CAs as well.
- For instance, the root CAs in the US are VeriSign, Thawte, and the US Postal Service. In such cases, there is no *single* root CA, which can be trusted by all the concerned parties.
- In our example, why should user A—a Japanese national, trust user B's root CA—a US-based organization?
- Cross-certification allows CAs and end users from different PKI domains to interact called **cross certification.**



Certificate Revocation:

Reasons for revocation:

- If the private key corresponding to the public key is stolen.
- The CA realizes that it had made mistake while issuing the certificate.
- The certificate holder leaves a job and the certificate was issued specifically while the Person was employed in that job.
- It checks:
  - Online revocation status
  - Off-line revocation status

Private Key Management:

To protect the private key by means:-

- Password protection
- Tokens
- Biometrics
- Smart Cards
- Apart from these, the private key used for digital signing must be destroyed. In contrast, the Private key used for encryption/decryption must be archived.
  - In case of certificate expiration, the user needs to update its key.
  - The CA should maintain history of certificates & keys to prevent any legal problems.

The PKIX (Public Key Infrastructure X.509) model:

(a) **Registration:**
In this process the end-entity (subject/user) registers to a CA. Usually this is via an RA.

(b) **Initialization:**
Process to verify that the end-entity is talking to the right CA.

(c) **Certification:**
In this step, the CA creates a digital certificate for the end-entity and returns it to the end-entity and keeps a copy for its own records.

(d) **Key-Pair Recovery:**
Keys used for encryption of some old documents may be required to be recovering data for decrypting. Key archival and recovery services can be provided by a CA.

(e) **Key Generation:**
PKIX specifies that the end-entity should be able to generate private-and public-key pairs, or the CA/RA should be able to do this for the end-entity.

(f) **Key Update:**
This allows issuing new key pair from old one by the automatic renewal of digital certificates. But there is a provision for issuing digital certificate manually.

(g) **Cross-certification:**
In this, each end-entity that are certified by different CAs can cross-verify each other.

(h) **Revocation:**
PKIX provides support for the checking of the certificate status in two modes: online or offline.

PKIX Architectural Model:

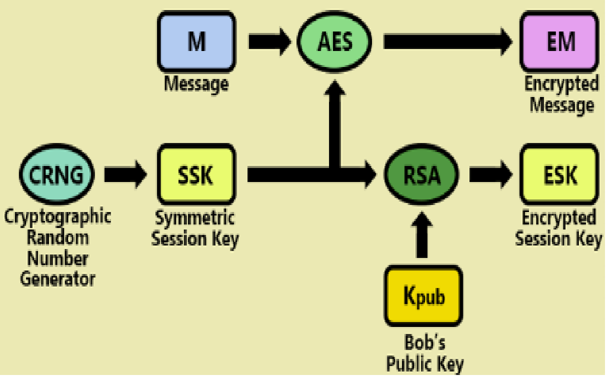
- **X.509 v3 Certificate & v2 Certificate Revocation List profiles**:
  Lists the use of various options while describing extensions of a digital certificate.
- **Operational Protocol**:
  Defines the underlying protocols that provide the transport mechanism.
- **Management Protocol**:
  Enables exchange of information between the various PKI entities and specifies the structure & details of PKI messages.
- **Policy outlines**:
  Defines policies for the creation of Certificate Policies & Certificate Practice Statements.
- **Timestamp & Data Certification Services**:
  Both are the trusted third parties that provide services to guarantee the existence of certificate & DCS verifies the correctness of data that it receives.

PKCS (Public Key Cryptography Standards)

Standard	**Description**
PKCS#1:	RSA Encryption Standard. Defines rules for calculating digital certificate.
PKCS#2:	RSA Encryption Standard for Message Digest.
PKCS#3:	Diffie-Hellman Key Agreement Standard.
PKCS#4:	NA. Merged with PKCS#1
PKCS#5:	Password Based Encryption(PBE). Defines method to encrypt symmetric key.
PKCS#6:	Extended Certificate Syntax Standard. Defines syntax for extending the basic attribute of an X.509 digital certificate.
PKCS#7:	Cryptographic Message Syntax Standard.
PKCS#8:	Private Key Information Standard.
PKCS#9:	Selected Attribute Types. Defines selected attribute for use in PKCS#6 extended certificates.
PKCS#10:	Certificate Request Syntax Standard
PKCS#11:	Cryptographic Token Interface Standard.
PKCS#12:	Personal Information Exchange Syntax Standard.
PKCS#13	Elliptic Curve Cryptography Standard.
PKCS#14	Pseudo –Random Number Generation Standard.
PKCS#15	Cryptographic Token Information Syntax standard.

Digital Envelop:

- A digital envelope is a secure electronic data container that is used to protect a message through encryption and data authentication.
- A Digital Envelope is created by symmetric key algorithm (e.g. AES) and the symmetric key.
- The symmetric key is then encrypted with an asymmetric key algorithm (e.g. RSA) and the recipient's public key.



UNIT-6-User Authentication

Authentication Basics:

What is Authentication…?:

- Proof of identity or we can say that "who is Who".
- It is the process of giving someone identity so that he or she can access that particular application or data.
- For e.g.: giving identity-card to a student of an institute.
- Authentication is the first step in any cryptographic solution
  - –Because unless we know who is communicating, there is no point in encryption what is being communicated.
- Authentication is any process by which a system verifies the identity of a user who wishes to access it.
- Establish trust before communication takes place.

Passwords:

•A password is a string of alphabets, numbers and special characters, which is supposed to be known only to the entity (usually person) that is being authenticated.

•Password Based Authentication

–Clear Text Passwords is the Simplest Password based Authentication Mechanism.

•How it works?

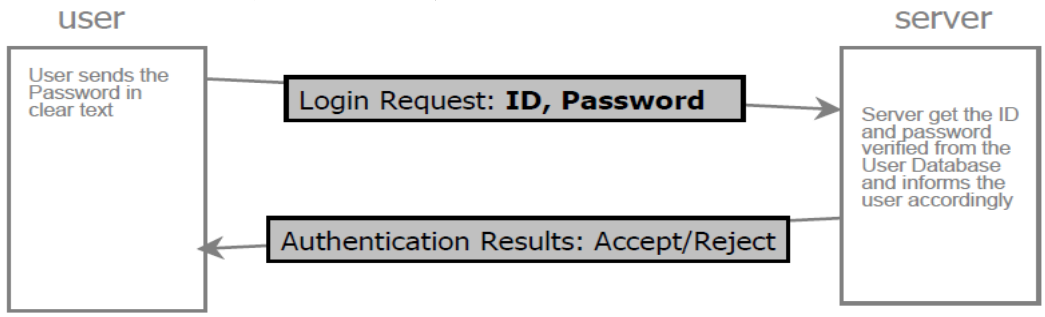
–Prompt for user ID and Password

–User enters user ID and Password

–User ID and Password Validation i.e user-id and password are validated.

–Authentication Result: Inform user accordingly.

Password Based Authentication:


user
server
User sends the Password in clear text
Login Request: ID, Password
Server get the ID and password verified from the User Database and informs the user accordingly
Authentication Results: Accept/Reject

User Authentication using Clear Text Password

Problems with Clear Text/plain-text Passwords:

–Database contains Passwords in clear text

•It is advised that password should not be stored in clear text in databases.

•The passwords should be stored in encrypted form in database.

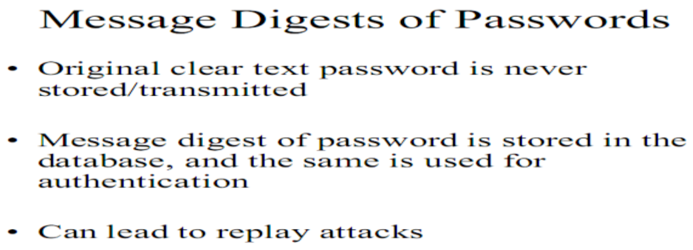
–Password travel in clear text/ plain-text from user's computer to the server

•If the attacker breaks into the communication link, he can easily obtain the clear text password.

•Something Derived from Password

–Message Digests of the Passwords

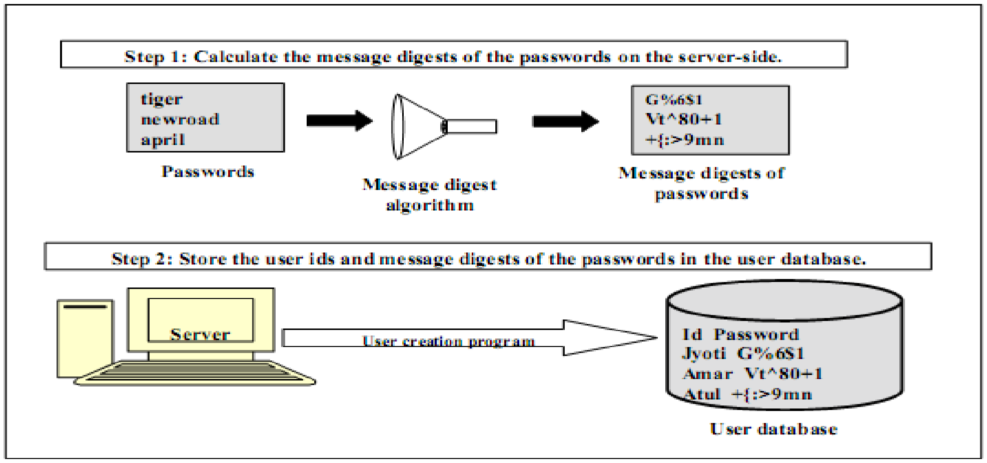
•Storing Message Digests as derived passwords in the user database.



Message Digests of Passwords

- Original clear text password is never stored/transmitted
- Message digest of password is stored in the database, and the same is used for authentication
- Can lead to replay attacks

Message Digests of Passwords


Step 1: Calculate the message digests of the passwords on the server-side.

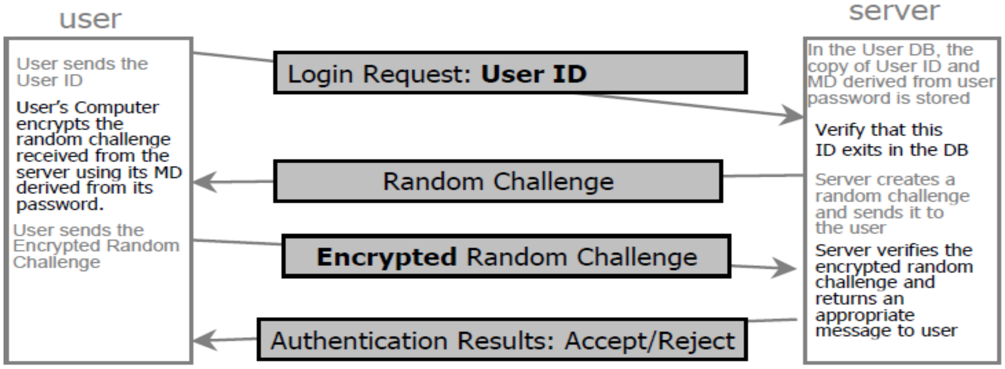
Step 2: Store the user ids and message digests of the passwords in the user database.

•Adding Randomness

–To improve the security and to detect a replay attack we need to add a bit of randomness to the earlier schemes

–Steps

1. Storing Message Digests as derived passwords in the user database.
2. User sends a login request
3. Server creates a random Challenge
4. User Signs the Random Challenge with the Message Digest of the Password
5. Server Verifies the Encrypted Random Challenge from the user
6. Server returns an appropriate message back to the user


User sends the User ID

User's Computer encrypts the random challenge received from the server using its MD derived from its password.

User sends the Encrypted Random Challenge

In the User DB, the copy of User ID and MD derived from user password is stored

Verify that this ID exits in the DB

Server creates a random challenge and sends it to the user

Server verifies the encrypted random challenge and returns an appropriate message to user
user
server
Login Request: User ID
Random Challenge
Encrypted Random Challenge
Authentication Results: Accept/Reject

Adding Randomness in Password Based Authentication

Authentication Tokens:

•It is an extremely useful alternative to a password

•These small devices are usually of the size of a small key chain.

•Usually an authentication Token has the following features

–Processor

–LCD for displaying outputs

–Battery

–Optionally a small keypad for entering information

–Optionally a real-time clock

•Each Authentication Token is pre-programmed with a unique number called as a random seed or just seed.

•Step Involved in Authentication Token:

1. **Creation of a Token**

–Created by the Authentication servers that are designed to use with authentication tokens.

–A unique value i.e. a seed is automatically placed or pre-programmed inside each token by the server.

–Server also keeps a copy of the seed against the user ID in the user database.

–Seed can be conceptually considered as a user password.

–Difference is that the user password is known to the user, seed value remains unknown to the user.

2. Use of the Token

–An Authentication Token automatically generates pseudorandom numbers called **one- time passwords**.

–One time passwords are generated randomly by authentication tokens using seed value.

–When a user wants to be authenticated by any server, the user will get a screen to enter user ID and the latest one-time password.

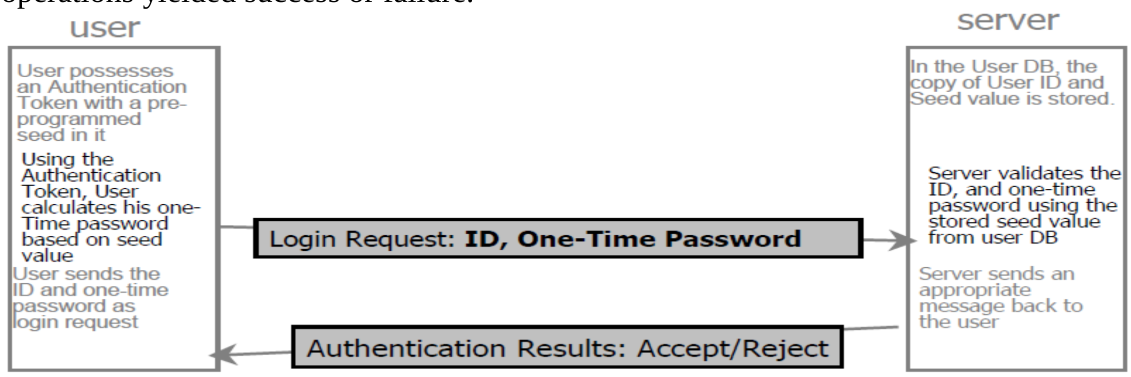
–The users enters its ID and gets is latest one-time password from the authentication token.

–The user ID and password travels to the server as a part of the login request

–Server verifies using some mechanism that this one-time password is created using the valid seed value.

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.


user
User possesses an Authentication Token with a pre-programmed seed in it
Using the Authentication Token, User calculates his one-Time password based on seed value
User sends the ID and one-time password as login request
server
In the User DB, the copy of User ID and Seed value is stored.
Server validates the ID, and one-time password using the stored seed value from user DB
Server sends an appropriate message back to the user
Login Request: ID, One-Time Password
Authentication Results: Accept/Reject

Authentication Tokens

Authentication Token Types:

1. Challenge/Response Tokens

2. Time-based Tokens

1. Challenge/Response Tokens:

Step 1: User Sends a Login Request.

In this technique, the user sends the login request only with his/her user id (and not the one-time password).

Step 2: Server Creates a Random Challenge

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.

Step 3: User Signs the Random Challenge with the Message Digest of the Password

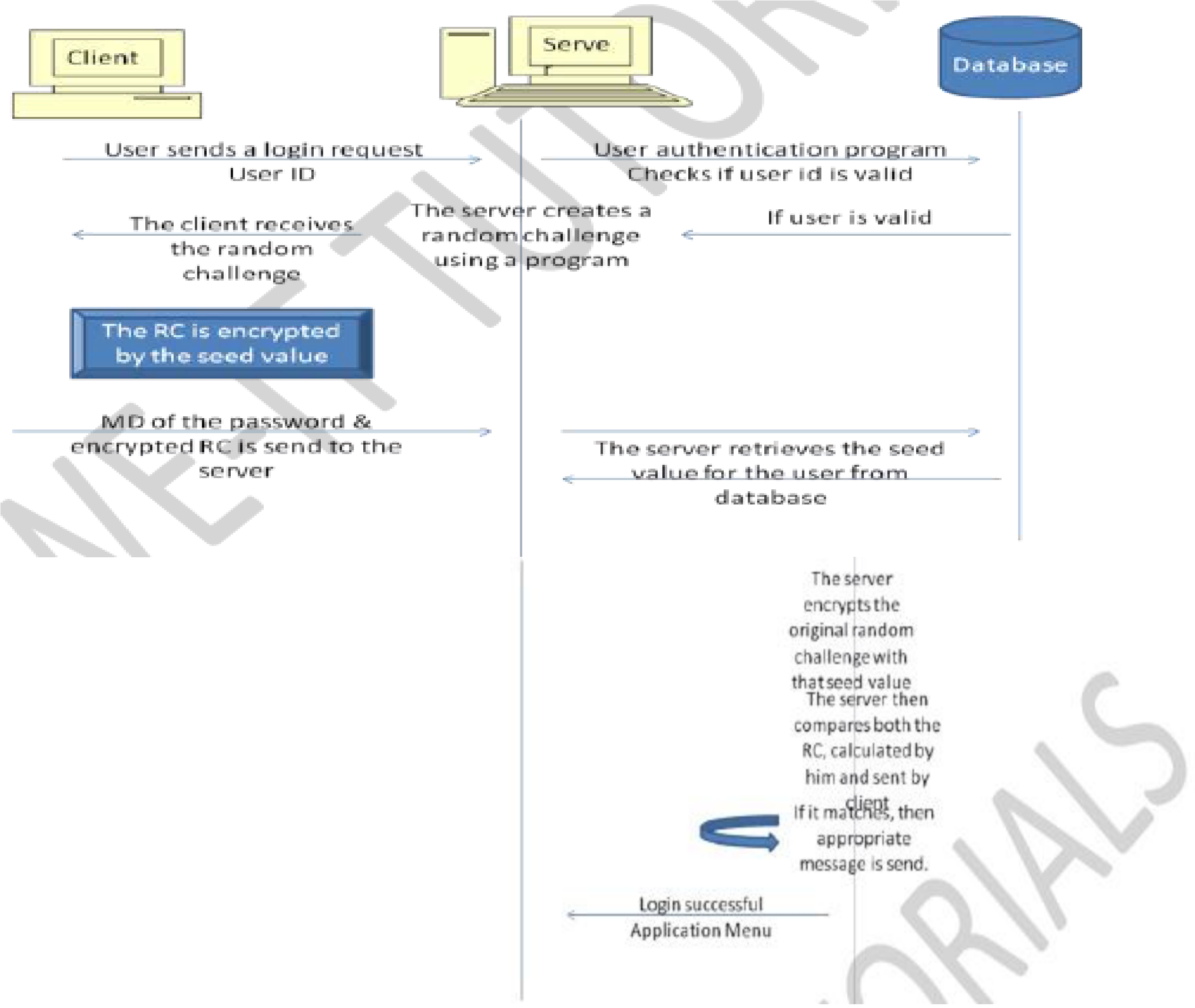
This request is then sent to the server as the login request.

Step 4: 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, the server must perform an identical operation.

Step 5: Server Returns an Appropriate Message Back to the User

Finally, the server sends an appropriate message back to the user, depending on whether the operation is success or failure.



Client
Serve
Database
User sends a login request
User ID
User authentication program
Checks if user id is valid
The client receives the random challenge
The server creates a random challenge using a program
If user is valid
The RC is encrypted by the seed value
MD of the password & encrypted RC is send to the server
The server retrieves the seed value for the user from database
The server encrypts the original random challenge with that seed value
The server then compares both the RC, calculated by him and sent by client
If it matches, then appropriate message is send.
Login successful
Application Menu

2. Time-based Tokens:

Step 1: **Password Generation and Login Request:**

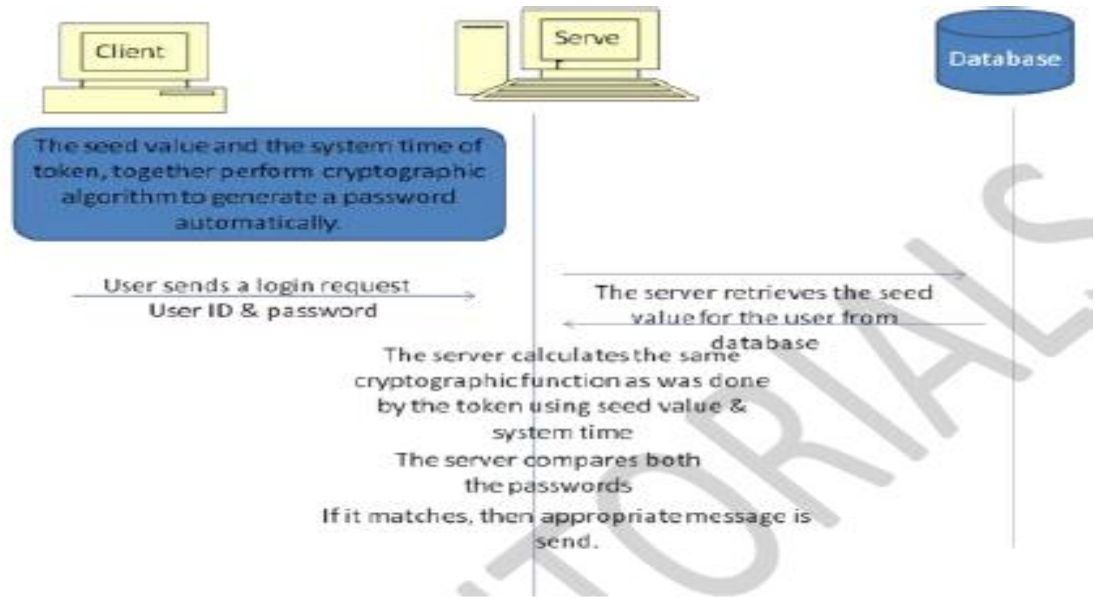
The seed value and the system time of token, together perform cryptographic algorithm to generate a password automatically.

Step 2: **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.

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 operation is success or failure.


Client
Serve
Database
The seed value and the system time of token, together perform cryptographic algorithm to generate a password automatically.
User sends a login request
User ID & password
The server retrieves the seed value for the user from database
The server calculates the same cryptographic function as was done by the token using seed value & system time
The server compares both the passwords
If it matches, then appropriate message is send.

Certificate Based Authentication:

•This is based on the Digital Certificates of the user.
•In PKI, the digital certificates are used for secure digital transactions.
•This can be re-used for user authentication as well.
•This is a stronger mechanism as compared to password based authentication

How does Certificate Based Authentication works?

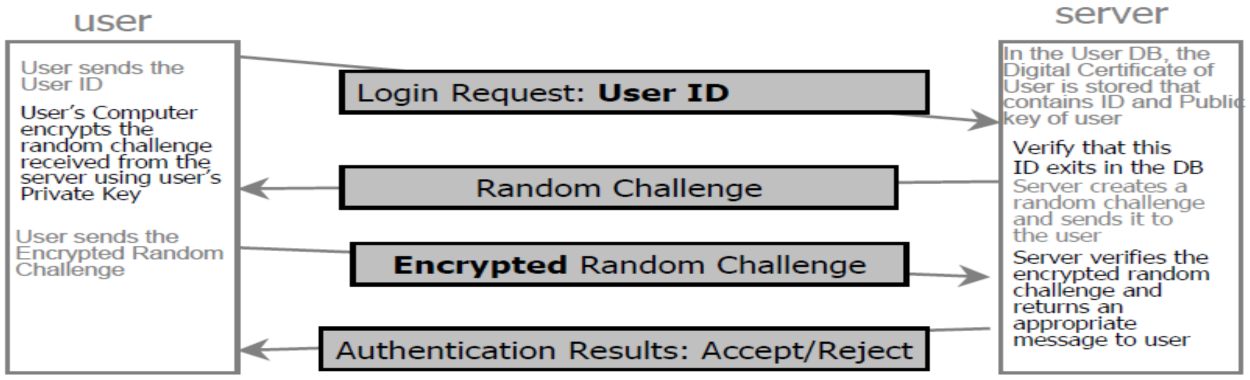
1. Creation, Storage and Distribution of Digital Certificates.
–Certificates are created by CA (Certificate Authority), sent to user as well as a copy to the server.

2. Login Request
–User sends its ID only.

3. Server Creates a Random Challenge
–User ID validity is checked.
–Sends random challenge in plain text to user.

4. User Signs the Random Challenge
–User signs the random challenge received from Server by using its Private Key
–User's private key is stored in a file in user computer
–To access its private key file, user has to give a correct password
–User sends the signed random challenge to the server

5. Server returns an appropriate message back to the user


user
server
User sends the User ID
User's Computer encrypts the random challenge received from the server using user's Private Key
User sends the Encrypted Random Challenge
Login Request: User ID
Random Challenge
Encrypted Random Challenge
Authentication Results: Accept/Reject
In the User DB, the Digital Certificate of User is stored that contains ID and Public key of user
Verify that this ID exits in the DB
Server creates a random challenge and sends it to the user
Server verifies the encrypted random challenge and returns an appropriate message to user

Certificate Based Authentication

Smart Cards:

- A smart card is a security token that has an embedded chip.
- Smart cards are typically the same size as a driver's license and can be made out of metal or plastic.
- They connect to a reader either by direct physical contact (also known as chip and dip) or through a short-range wireless connectivity standard such as Near Field Communication (NFC).
- It is Portable.
- Used to perform cryptographic mechanisms.

Use of Smart Cards:

•The use of Smart Cards is related to Certificate Based Authentication

•This is because the smart cards allows the generation of public-private key pairs within the card

•They also support the storage of digital certificates within the card.

•The private key always remain in the smart card in a secure fashion

•The public key and the certificate is exposed outside

•Also the smart cards are capable of performing cryptographic functions such as encryption, decryption, message digest creation and signing within the card

–Thus during the certificate based authentication, the signing of random challenge sent by the server can be performed inside the card

Problems and issues in Smart Cards:

•Lack of standardization and inter-operability between smart cards vendors

•Smart card reader are not yet a part of a desktop computer like hard disk drive or floppy drives

•Non-availability of smart card reader driver software

•Non-availability of smart card aware cryptographic service software

•Cost of smart cards and card reader is high

Biometric Authentication:

Definition:

Biometrics refers to the automatic identification of a person based on his or her physiological or behavioral characteristics.

•A biometric device works on the basis of some human characteristics, such as fingerprints, voice or the pattern of lines in the iris of your eye

•The user database contains a sample of user's biometric characteristics

•During the authentication, the user is required to provide another sample of the users' biometric characteristic.

•This is matched with the one in the database, and if the two samples are same, the user is considered to be a valid one.

•The samples produced during every authentication process can vary slightly. (e.g. cuts on the finger)

•An approximate match can be acceptable

•Any Biometric Authentication System defines two configurable parameters:

–False Accept Ratio (FAR):

•It is a measurement of the chance that a user who should be rejected is actually accepted by a system as good enough

–False Reject Ratio (FRR):

•It is a measurement of the chance that a user who should be accepted as valid is actually rejected by a system as not good enough

•Thus FAR and FRR are exactly opposite to each other.

Biometric characteristics:

1) Physiological
2) Behavioral

Physical biometrics:

- Fingerprint
- Facial recognition/face location
- Hand geometry
- Iris scan
- Retina scan

Fingerprint recognition

- A live acquisition of a person's fingerprint.
- Dots (very small ridges),
- Space between two temporarily divergent ridges),
- Spurs (a notch protruding from a ridge),
- Bridges (small ridges joining two longer adjacent ridges), crossovers (two ridges that cross each other).

Facial Recognition

1. Capture image
2. Find face in image
3. Extract features (store template)
4. Compare templates
5. Declare matches

Hand Geometry

Hand or finger geometry is an automated measurement of many dimensions of the hand and fingers.

Iris recognition

Iris scanning measures the iris pattern in the colored part of the eye.

Retina recognition

Images back of the eye and compares blood vessels with existing data.

<u>Behavioral biometrics</u>

- Speaker/ voice recognition.
- Signature/ handwriting.
- Keystroke/ patterning.

Speaker / Voice Recognition

- Voice or speaker recognition uses vocal characteristics to identify individuals using a pass-phrase.
- A telephone or microphone can serve as a sensor.

Signature Verification

- An automated method of measuring an individual's signature.
- This technology examines speed, direction, and pressure of writing; the time that the stylus is in and out of contact with the "paper''.

Keystroke dynamics

- It is an automated method of examining an individual's keystrokes on a keyboard.
- This technology examines such dynamics as speed and pressure, the total time taken to type particular words, and the time elapsed between hitting certain keys.

APPLICATIONS:

- Prevent unauthorized access to ATMs, Cellular phones Desktop PCs.
- Criminal identification.
- In automobiles biometrics can replace keys with keyless entry devices.
- Airport security.

UNIT:7-NETWORK SECURITY AND VPN

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

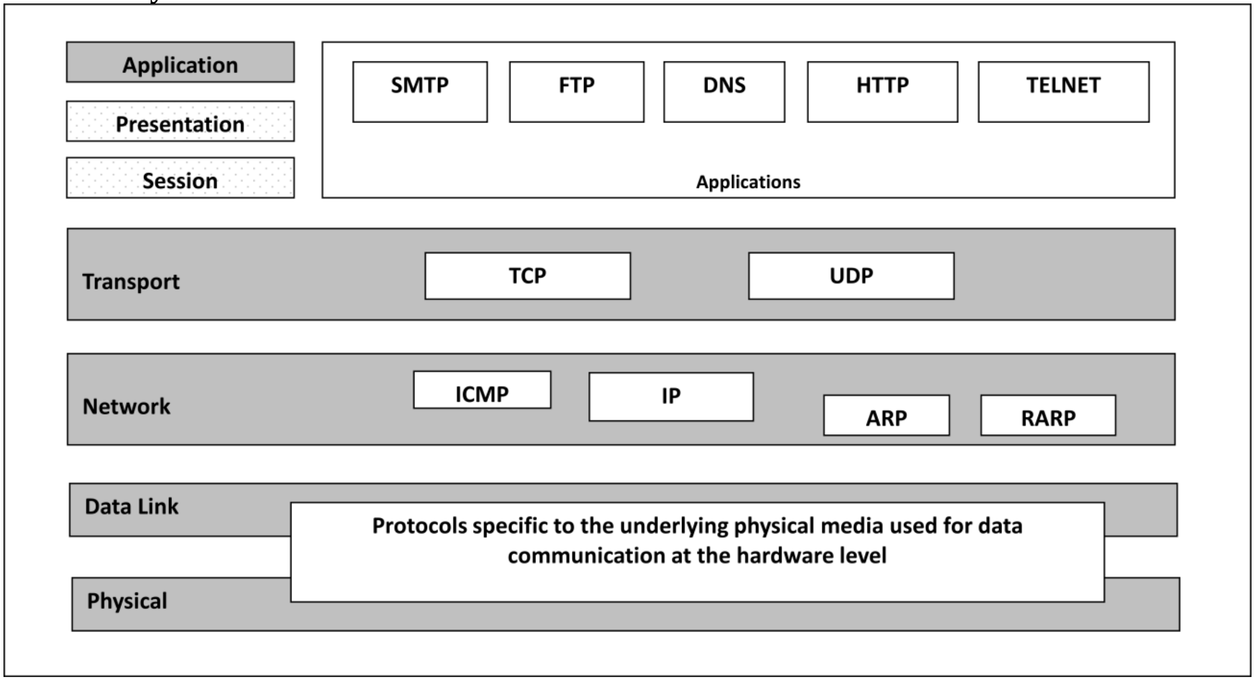
……………………………………………………………………………………………...

TCP/IP:

TCP/IP Protocol Suite:

- The TCP/IP protocol suite was developed prior to the OSI model. Therefore, the layers in the TCP/IP protocol suite do not exactly match those in the OSI model.
- **TCP/IP protocol suite is made of five layers: Application Layer, Transport Layer, Internet Layer, Network Access Layer**
- *TCP/IP* is a hierarchical protocol made up of interactive modules, each of which provides a specific functionality; however, the modules are not necessarily interdependent.
- At the transport layer, *TCP/IP* defines three protocols: **Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and Stream Control Transmission Protocol (SCTP).**
- At the Internet layer, the main protocol defined by TCP/IP is the **Internet Protocol (IP);** there are also some other protocols that support data movement in this layer.

TCP/IP Layers:



TCP segment format:

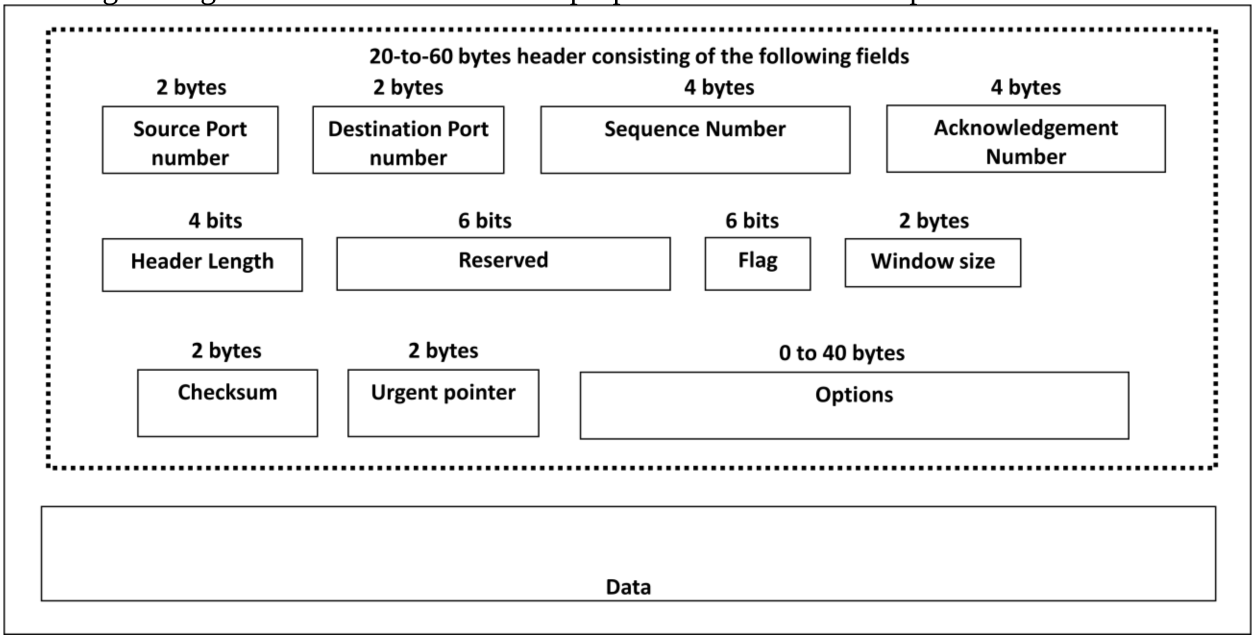
A packet in TCP is called a **segment.** The segment consists of a header of 20 to 60 bytes, followed by data from the application program. The header is 20 bytes if there are no options and up to 60 bytes if it contains options.

Source port address:

This is a 16-bit field that defines the port number of the application program in the host that is sending the segment. This serves the same purpose as the source port address in the UDP.

Destination port address:

This is a 16-bit field that defines the port number of the application program in the host that is receiving the segment. This serves the same purpose as the destination port address in the UDP.



Sequence number:

This 32-bit field defines the number assigned to the first byte of data contained in this segment. As TCP is a stream transport protocol. To ensure connectivity, each byte to be transmitted is numbered. The sequence number tells the destination which byte in this sequence is the first byte in the segment. During connection establishment each party uses a random number generator to create an **initial sequence number** (ISN), which is usually different in each direction.

Acknowledgment number:

This 32-bit field defines the byte number that the receiver of the segment is expecting to receive from the other party. If the receiver of the segment has successfully received byte number *x* from the other party, it Returns *x*+1 as the acknowledgment number.

Header length:

This 4-bit field indicates the number of 4-byte words in the TCP header. The length of the header can be between 20 and 60 bytes. Therefore, the value of this field is always between 5 (5 *4=20) and 15 (15*4=60).

Reserved: This is a 6-bit field reserved for future use.

Control:

This field defines 6 different control bits or flags . One or more of these bits can be set at a time. These bits enable flow control, connection establishment and termination, connection abortion, and the mode of Flags from left to right:

Window size:

This field defines the window size of the sending TCP in bytes. Note that the length of this field is 16 bits, which means that the maximum size of the window is 65,535 bytes.

Checksum:

The 16-bit checksum field is used for error-checking of the header and data.

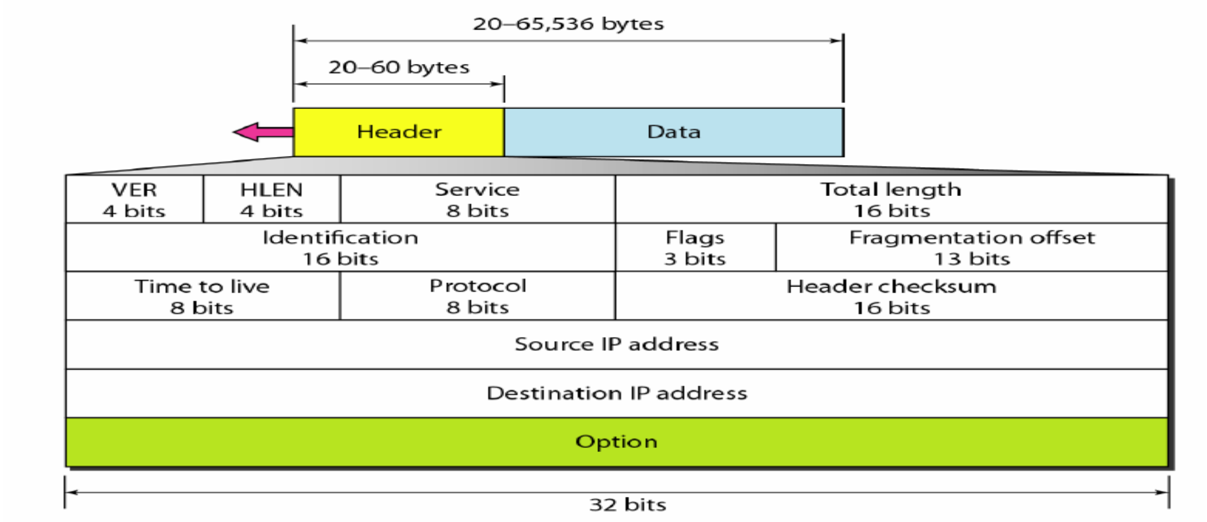
Urgent pointer:

if the URG flag is set, then this 16-bit field is an offset from the sequence number indicating the last urgent data byte.

IP DATAGRAM FORMAT:

- Packets in the network (internet) layer are called *datagram*.
- A datagram is a variable-length packet consisting of two parts: header and data.
- The header is 20 to 60 bytes in length and contains information essential to routing and delivery.

IP header format:



Version (VER):

This 4-bit field defines the version of the IP protocol. Currently the version is 4(IPv4).

Header length (HLEN):

This 4-bit field defines the total length of the datagram header in 4-byte words. This field is needed because the length of the header is variable (between 20 and 60 bytes). When there are no options, the header length is 20 bytes, When the option field is at its maximum size(i.e. 60)

Service type (TOS):

It defines how the datagram should be handled. Part of the field was used to define the precedence of the datagram; the rest defined the type of service (low delay, high throughput, and so on).

Total length:

It defines the total length of the datagram including the header in bytes. It is a 16-bit number so maximum IP size is 216 bytes.

Identification:

This 16-bit field identifies a datagram originating from the source host. The combination of the identification and source IP address must uniquely define a datagram as it leaves the source host.

Flags:

This is a three-bit field. The first bit is reserved (not used). The second bit is called the do not fragment bit. If its value is 1, the machine must not fragment the datagram. If its value is 0, the datagram can be fragmented if necessary. The third bit is called the more fragment bit. If its value is 1, it means the datagram is not the last fragment; there are more fragments after this one. If its value is 0, it means this is the last or only fragment.

Fragmentation offset:

This 13-bit field shows the relative position of this fragment with respect to the whole datagram.

Time to live:

A datagram has a limited lifetime in its travel through an internet. This field was originally designed to hold a timestamp, which was decremented by each visited router. The datagram was discarded when the value became zero.

Protocol:

This 8-bit field defines the higher-level protocol that uses the services of the IP layer. An IP datagram can encapsulate data from several higher level protocols such as TCP, UDP, ICMP, and IGMP. This field specifies the final destination protocol to which the IP datagram should be delivered.

Header Checksum:

This fields represents a value that is calculated using an algorithm covering all the fields in header. This field is used to check the integrity of an IP datagram.

Source address:

This 32-bit field defines the IP address of the source. This field must remain unchanged during the time the IP datagram travels from the source host to the destination host.

Destination address:

This 32-bit field defines the IP address of the destination. This field must remain unchanged during the time the IP datagram travels from the source host to the destination host.

Firewall:

Firewalls can be used to protect a local system or network of systems (Internal Network) from Out-side networks (Internet) from security threats.

- Special type of router.
- Frequently used to prevent unauthorized internet users from accessing private networks connected to the internet, especially intranets.
- Controls transmission between internal and external networks. i.e. 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.
- It is essentially a barrier between two networks that evaluates all incoming or outgoing traffic to determine whether or not it should be permitted to pass to the other network. i.e. decides what to allow/disallow.
- Can be implemented in both hardware and software, or a combination of both.
- At broad level, there are two kind of attacks:
  - Most corporations have large amounts of valuable and confidential data in their networks. Leaking of this critical information to competitors can be a great setback.
  - Apart from the danger of the insider information leaking out, there is a great danger of the outside elements (such as viruses and worms) entering a corporate network to create disaster.

Firewall characteristics/ Design Goals of Firewalls:

A firewall is defined as collection of components placed between two networks that collectively have Following characteristics:

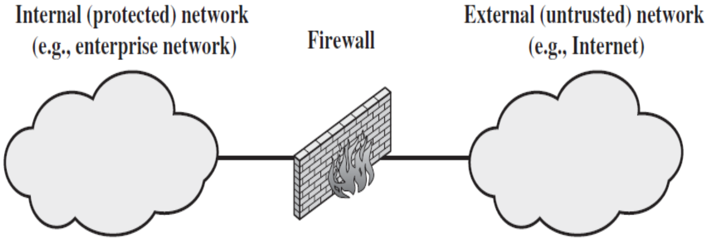
1. All traffic from inside to outside, and vice versa, must pass through the firewall.

   – This is achieved by physically blocking all access to the local network except via the firewall.

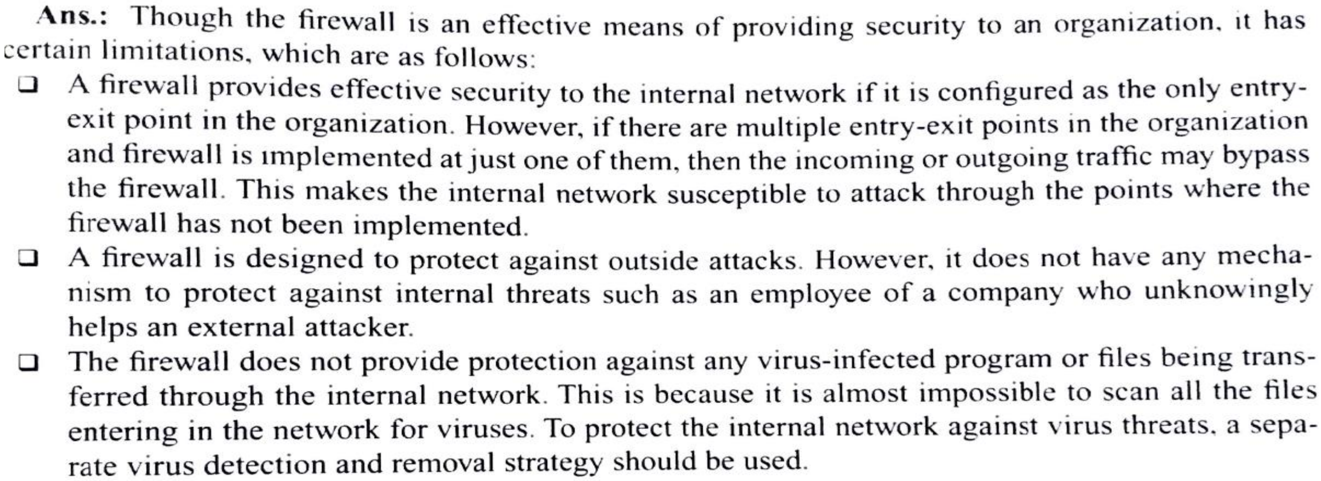
2. Only authorized traffic, as defined by the local security policy, will be allowed to pass.

– Various types of firewalls are used, which implement various types of security policies.

3. The firewall itself must be strong enough, so as to render attacks on it useless.


Internal (protected) network
(e.g., enterprise network)
Firewall
External (untrusted) network
(e.g., Internet)

Limitations of Firewalls:

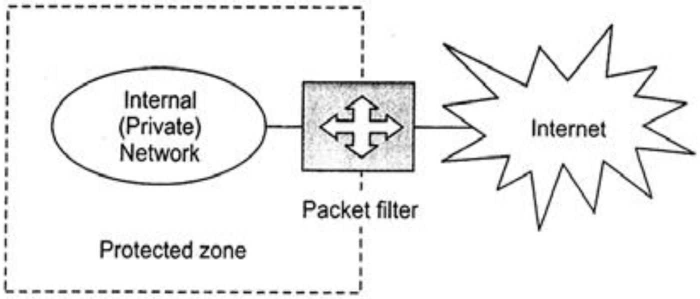


Ans.: Though the firewall is an effective means of providing security to an organization, it has certain limitations, which are as follows:

- ❑ A firewall provides effective security to the internal network if it is configured as the only entry-exit point in the organization. However, if there are multiple entry-exit points in the organization and firewall is implemented at just one of them, then the incoming or outgoing traffic may bypass the firewall. This makes the internal network susceptible to attack through the points where the firewall has not been implemented.
- ❑ A firewall is designed to protect against outside attacks. However, it does not have any mechanism to protect against internal threats such as an employee of a company who unknowingly helps an external attacker.
- ❑ The firewall does not provide protection against any virus-infected program or files being transferred through the internal network. This is because it is almost impossible to scan all the files entering in the network for viruses. To protect the internal network against virus threats, a separate virus detection and removal strategy should be used.

Types of Firewalls:

- ➢ Packet Filters.
- ➢ Application Level Filtering.
- ➢ Circuit Level Gateways.

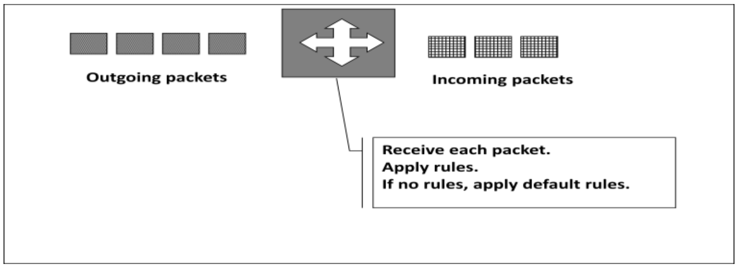


Packet Filtering Firewall:

- A firewall may act as a packet filter.
- It can act as a positive filter, if pass only packets that meet specific criteria, or as a negative filter, if rejecting any packet that not meets certain criteria.
- A packet filtering firewall applies a set of rules to each incoming and outgoing IP packet and then forwards or discards the packet.
- It is also called as screening router or screening filter.

- The idea of a packet filter is shown in figure

A packet filter performs the following functions/operations:



Conceptually, a packet filter can be considered as a router that perform 3 main actions, that is shown in figure

(a) Receive each packet as it arrives.

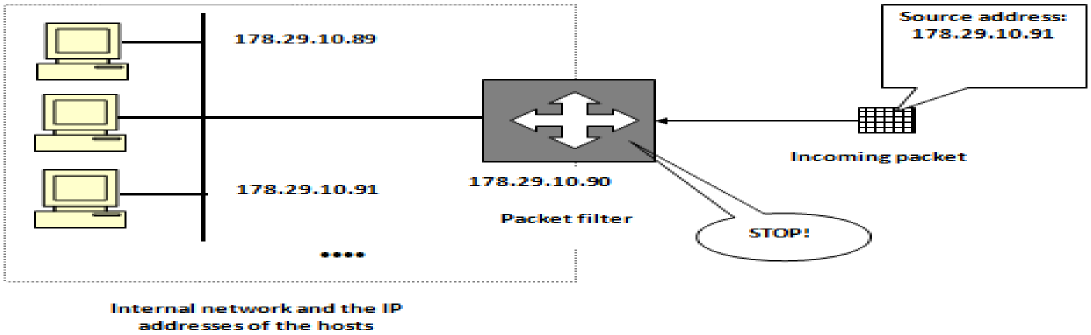
(b) Pass the packet through a set of rules, based on the contents of the IP and transport header fields of the packet. If there is a match with one of the set rules, decide whether to accept or discard the packet based on that rule.

For example: a rule could specify: disallow all incoming traffic from an IP address 157.29.19.10 (this IP address is taken just as an example)

(c) If there is no match with any rule, take the default action. The default can be discard all packets or accept all packets.

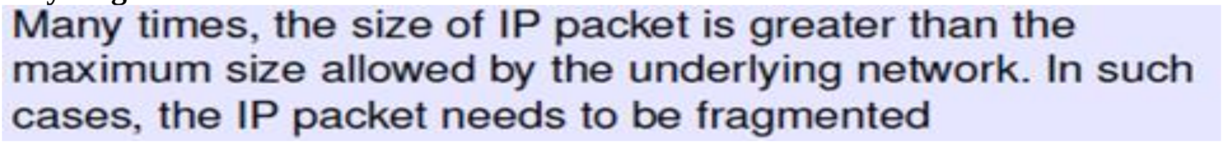
Attacks on Packet Filtering Firewall:

- **IP address spoofing:** The intruder transmits packets from the outside network towards internal network with a source IP address set equals to one of the IP address of an internal host/user.
  Countermeasure: is to discard all the packets that arrive at the incoming side of firewall, with a source IP address set equals to one of the IP address of an internal host/user.


178.29.10.89
178.29.10.91
178.29.10.90
Packet filter
Source address: 178.29.10.91
Incoming packet
STOP!
....
Internal network and the IP addresses of the hosts

- **Source routing attacks:** The source station specifies the route that a packet should take as it crosses the Internet, that this will bypass security measures that do not analyze the source routing information.
  Countermeasure: is to discard all packets that use this option.

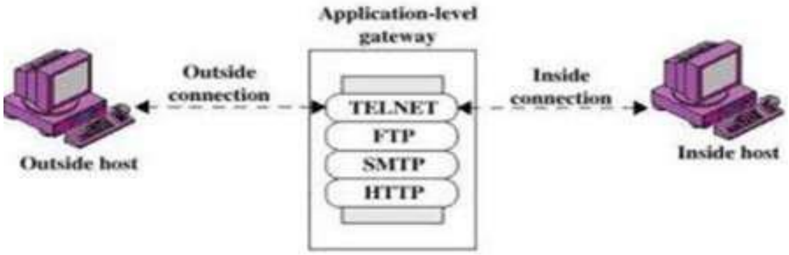
- **Tiny fragment attacks:**


Many times, the size of IP packet is greater than the maximum size allowed by the underlying network. In such cases, the IP packet needs to be fragmented

  Countermeasure: A tiny fragment attack can be defeated by enforce a rule that the first fragment of a packet must contain a predefined minimum amount of the transport header. If the first fragment is rejected, the filter can remember the packet and discard all subsequent fragments.

Application Level Gateway:

An application level firewall also called a proxy server or bastion host.. It operates at application layer of the OSI model. It handles the flow of application level traffic. It is as shown in fig.



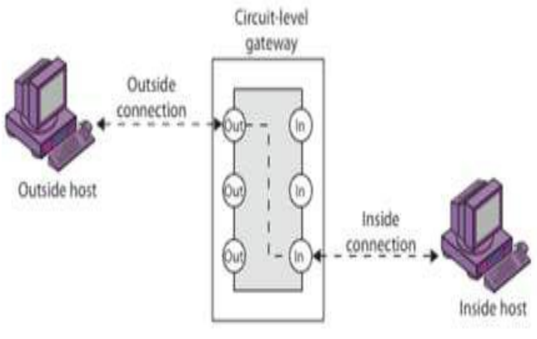
The operation of Application Level Gateway is as follow:

1. The user contacts the gateway using a TCP/IP application such as Telnet, HTTP or FTP.
2. In response, the gateway asks the user for the name, IP address and other information of the remote host to be accessed. It also asks the user to present its user ID and password to access the gateway.
3. When user provides valid user ID and authentication information to the gateway.

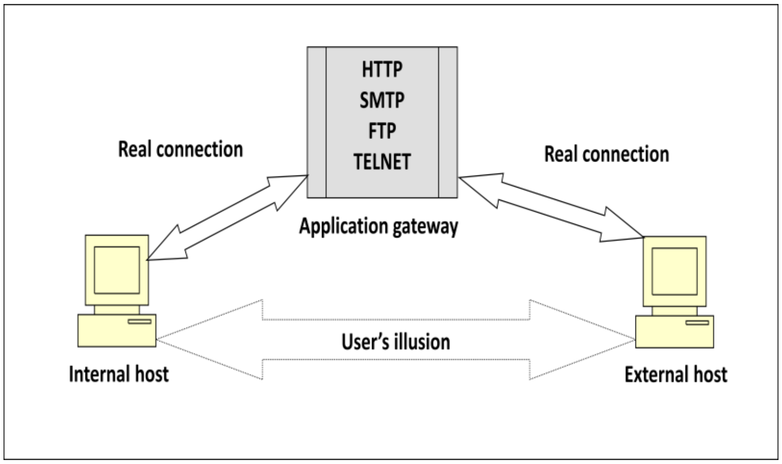
4. After verifying the user, the gateway contacts the application on the remote host on behalf of the user. Then relays TCP segment containing the application data between the two end points.
5. Now, the application gateway serves as a proxy of the original user and delivers application data in both directions, i.e. from remote host to user and vice-versa.
6. If the gateway does not implement the proxy code for a specific application, the service is not supported and cannot be forwarded across the firewall.

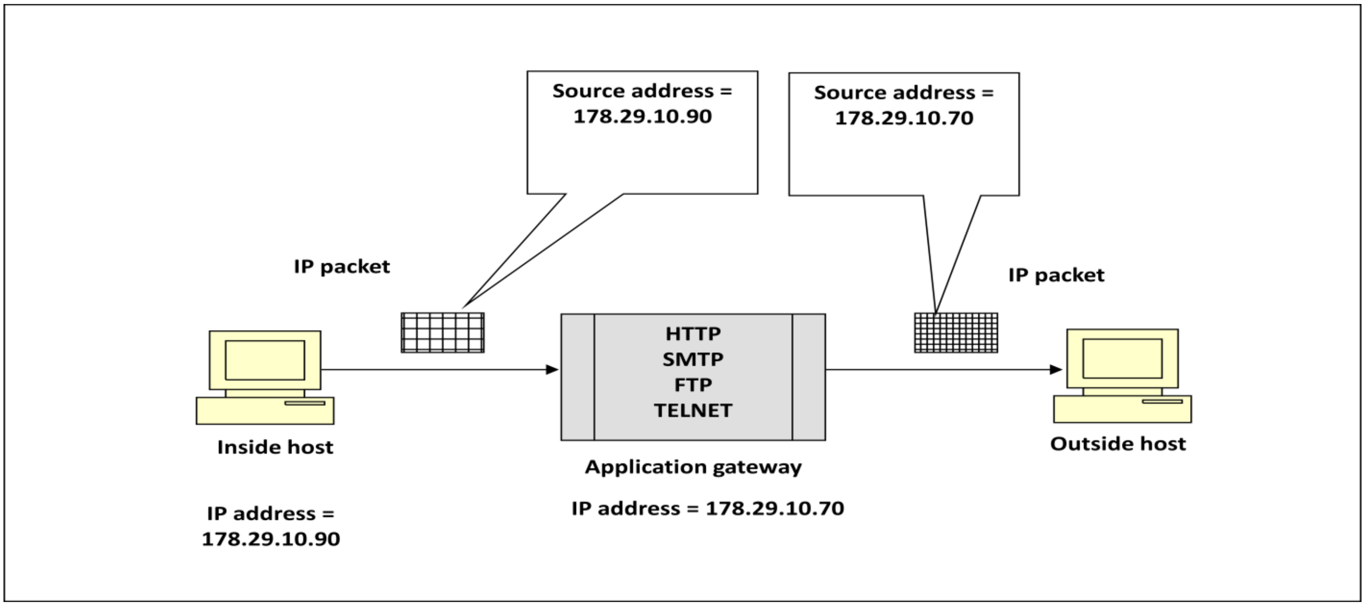
Circuit-Level Gateway –

1. It is a variation of the application gateway.
2. It works at the session layer of the OSI model or the TCP layer of TCP/IP.
3. This can be a stand-alone system or it can be specialized function performed by an Application level Gateway for certain applications.
4. A circuit level gateway does not permit an end-to-end TCP connection.
    - This gateway set up two TCP connections:
    - One between itself and a inner TCP user.
    - Second between itself and a outside TCP user (remote host).
5. The user is not aware of it and thinks that there is a direct connection between itself and the remote host.
6. Also, this gateway changes the source IP addresses in packets from the end user’s IP address to it’s own. This way, The IP addresses of the internal users are hidden from the outside world.
7. Once the two connections are established, the gateway typically relays TCP segment from one connection to the other without examining the contents.
8. The security function consists of determining which connections will be allowed.



(c) Circuit-level gateway

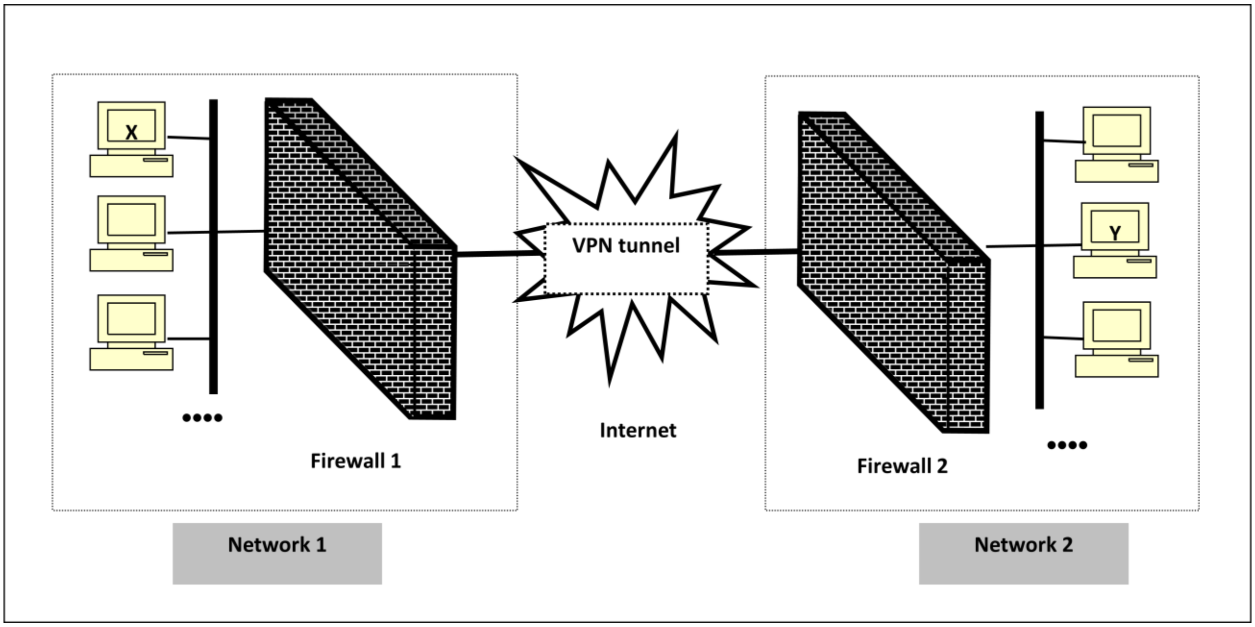




Virtual Private Network (VPN):

- A VPN is thus a mechanism to simulate a private network over a public network, such as the Internet.
- The term *virtual* signifies that it depends on the use of virtual connections.
- These connections are temporary and do not have any Physica1 presence. They are made up of packets.
- Uses the Internet as if it is a private network.
- Far less expensive than a leased line.
- Uses IPSec protocol.

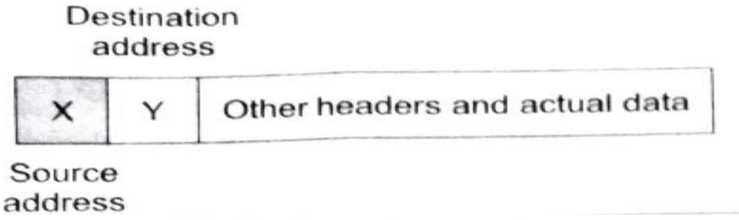
VPN Architecture:



- We have shown two networks, *Network* I and *Network* 2. Network l connects to the Internet via a firewall named Firewall I. Similarly, *Network* 2 connects to the Internet with its own firewall 2.
- The two firewalls are *virtually* connected to each other via the Internet. We have shown this with the help of a *VPN tunnel* between the two firewalls.

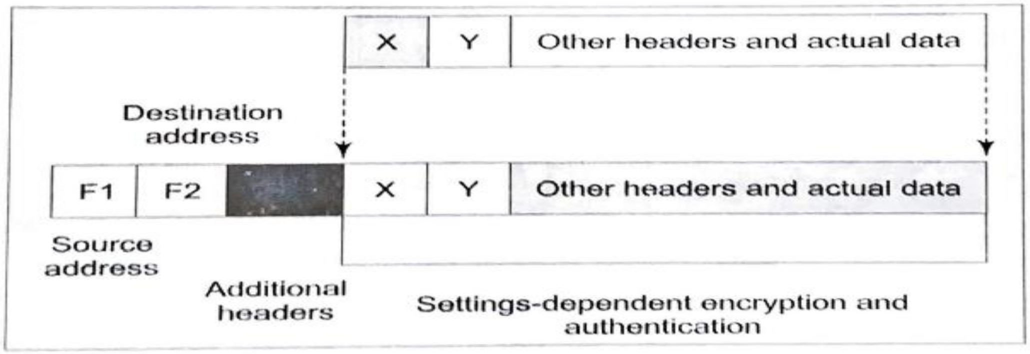
Let us understand how the VPN protects the traffic passing between any two hosts on the two different networks. For this, let us assume that host *X* on *Network* 1 wants to send a data packet to host Y on *Network* 2. This transmission would work as follows.

I. **H**ost *X* creates the packet, inserts its own IP address as the source address and the IP address of host *Y* as the destination address. This is shown in figure. It sends the packet using the appropriate mechanism.



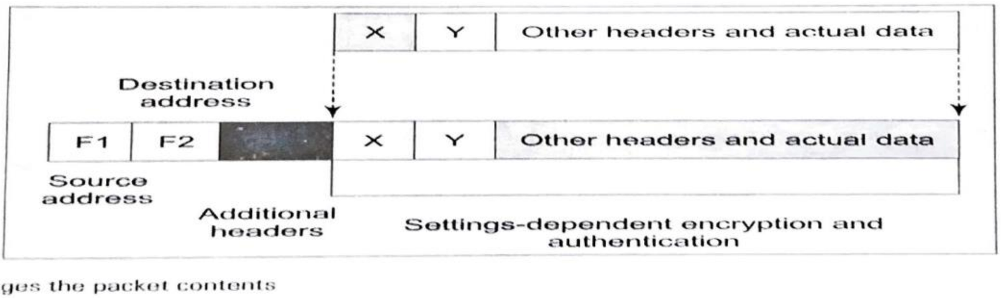
Original packet

2. The packet reaches firewall 1. As we know, firewall 1 now adds new headers to the packet. In these new headers, it changes the source IP address or the packet from that of host *X* to its own address (i.e. the IP address of *Firewall* 1, say F1). It also changes the destination IP address of the packet from that of host *Y* to the IP address of *Firewall* 2. say F2). This is shown in Fig. It also performs the packet encryption and authentication, depending on the settings and sends the modified packet over the Internet.


X
Y
Other headers and actual data
Destination address
F1
F2
X
Y
Other headers and actual data
Source address
Additional headers
Settings-dependent encryption and authentication

Firewall 1 changes the packet contents

3.The packet reaches *firewall1 2* over the internet, via one more routers, as usual, *Firewall 2* discards the outer header and performs the outer header and performs appropriates decryption and other cryptographic functions as necessary. This yields the original packets, as was created by host X in step 1. This is shown in fig. It then takes a look the plain text contents of the packets and realizes that the packet is meant for host Y. Therefore, it delivers the packet to host Y.


X
Y
Other headers and actual data
Destination address
F1
F2
X
Y
Other headers and actual data
Source address
Additional headers
Settings-dependent encryption and authentication

Firewall 1 changes the packet contents

Main Network Protocols:

There are three network protocols.

IPSec:(*Internet Protocol Security*): It is a framework for uses cryptographic security services developed by the IETF to protect secure exchange communications over Internet Protocol (IP).

PPTP(*Point-to-Point Tunneling Protocol*): It is a network protocol. It mainly support the vpn connectivity bet a single user and a LAN.

L2TP:(*Layer Two Tunneling Protocol*): It is an extension of the Point-to-Point Tunneling Protocol (PPTP) used by Internet service providers (ISPs) to operate Virtual Private Networks (VPNs).

IP Security (IPSec) Protocols:

- Before IPSec was initiated, the IP packets were prone to security failure.
- The technology that brings secure communications to the internet protocol layer or network layer is called IP Security, commonly abbreviated IPSec.
- IPSec is a set of services and protocols that provide a complete security solution for an IP networks.
- It is a collection of protocols designed by the Internet Engineering Task Force (IETF) to provide security in the internet layer.
- It can be used in protecting data flows between a pair of host(host-to-host), between a pair of security gateways(network-to-network), or between a security and a host(network-to-host).

General IP Security mechanisms:

It provides:

Authentication: Ensures that packets are arriving from the actual source.

Confidentiality: It allows two communicating nodes to transfer msg in an encrypted form in order to prevent third party.

Key management: Provide platform to key exchange in a secured manner.

Applications of IP security: (Important)

- IPSec provides the capability to secure communications across a LAN, across private and public WANs, and across the Internet. Examples of its use include:

Secure remote access over the Internet:

- An end user whose system is equipped with IP security protocols can make a local call to an Internet Service Provider (ISP) and gain secure access to a company network. This reduces travelling cost and time wastage of employees and telecommuters.

Secure branch office connectivity over the Internet:

- A company can build a secure virtual private network over the Internet or over a public WAN. This enables connecting all the branches of company. That will save the costs of creating a private network and network management overhead.

Establishing extranet and intranet connectivity with partners:

- IPSec can be used to secure communication with other organizations, ensuring authentication and confidentiality and providing a key exchange mechanism.

Enhancing electronic commerce security:

- Even though some Web and electronic commerce applications have built-in security protocols, the use of IPSec enhances that security.

Benefits of IP security: (Important)

- IPSec can be transparent to end users.
    - There is no need to train users on security mechanisms.
    - No need to issue or cancel keys to and from the users.
- When IPSec is implemented in a firewall or router, it provides strong security that can be applied to all traffic crossing the perimeter.
    - Traffic within a company or workgroup does not have to use IPSec, thus it minimize the overhead of security-related processing.
- IPSec in a firewall is resistant to bypass if all traffic from the outside must use IP and the firewall is the only means of entrance from the Internet into the organization.
- Since IPSec is implemented at network layer, there is no need to make any changes at the upper layers such as transport layer (TCP, UDP) and application layer.
- IPSec can provide security for individual users if needed. Individuals can set up a secure virtual sub-network within an organization for sensitive applications.

IP security services: (Important)

- IPSec provides security services at the IP layer.
- Two protocols are used to provide security:
    - An authentication protocol designated by the header of the protocol, **Authentication Header (AH).**
    - And a combined encryption/ authentication protocol designated by the format of the packet for that protocol, **Encapsulating Security Payload (ESP).**

- Lists the following services:
  1. Access control
  2. Connectionless integrity
  3. Data origin authentication
  4. Rejection of replayed packets (a form of partial sequence integrity)
  5. Confidentiality (encryption)
  6. Limited traffic flow confidentiality

IPSec Architecture/ Protocol:

The IPSec architecture comprises of different protocols like:
1. Authentication Header (AH) protocol.
2. Encapsulating Security Payload (ESP).

1. Authentication Header (AH) protocol:

- Provides support for data integrity and authentication (MAC code) of IP packets.
- Guards against replay attacks.

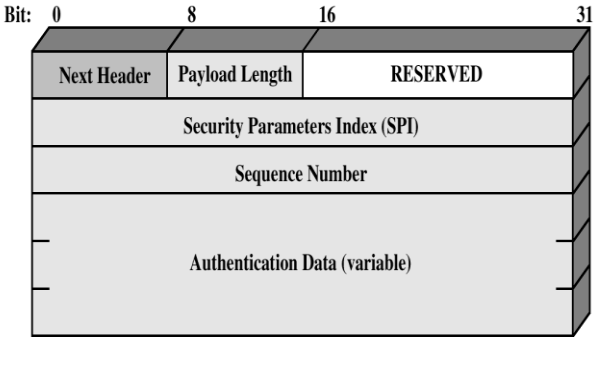

Bit: 0
8
16
31
Next Header
Payload Length
RESERVED
Security Parameters Index (SPI)
Sequence Number
Authentication Data (variable)

Figure 6.3 IPSec Authentication Header

A brief description each field:

Next header: The 8-bit next-header field defines the type of payload carried by the IP datagram

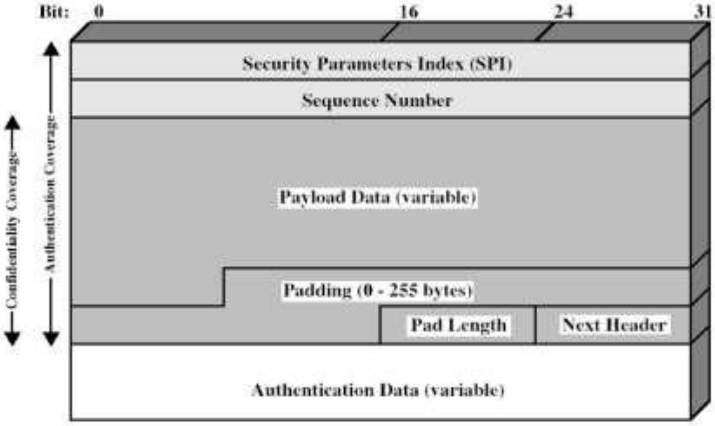
(such as TCP, UDP, ICMP).
Payload length: this is 8 bit field is misleading. It defines the length of the authentication header.

Security parameter index: this is 32bit security parameter index (SPI) field plays the role of a virtual-circuit identifier and is the same for all packets sent during a connection called a security association.

Sequence Number: a 32-bit sequence number provides ordering information for a sequence of datagram's. The sequence number is not repeated even if a packet is retransmitted.

Authentication data: finally, the authentication data field is the result of applying a hash function to the entire IP datagram except for the fields that are changed during transmit.

ESP protocol:
Due to the limitations of the authentication header IPSec defined an alternative protocol that provides source authentication and integrity and privacy called Encapsulating Security Payload (ESP).



ESP fields:
Security Parameter Index (SPI):
- It is a 32 bit field. It is used in combination with source and destination address

Sequence number
- It is a 32 bit field used to prevent replay attack.

Payload data:
- It is variable length field. It contains the transport layer segment or ip packet.

Padding:
- This field contains padding bits (if any).
- These bits are mainly used in encryption algorithm.

Pad length:
- It is an 8 bit field. It indicates the number of bytes padded in the previous field.
- It is reserved bits for next header.

Next header:
- It is an 8 bit field. It indicates the type of data content in the payload data field.

Authentication Data:
- It is a variable length field. It contains the ICV(integrity check value)