UNIT-I

CLASSICAL ENCRYPTION TECHNIQUES

(1) OSI Security Architecture:-

→ OSI Security Architecture is useful to managers as a way of organizing systematic approach.

→ ITU-T Recommendation X.800, security Architecture for OSI.

→ OSI Security Architecture defines a systematic way to

1. Defining the requirements for security.
2. Characterizing the approaches to satisfy those requirements.

→ OSI Security Architecture Contains:

Security Attack: An action that compromises the security of information owned by an organization (or a person).

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 data processing systems and information transfers.

1. Security Attacks:-

Security attacks are of two types

a. Passive attacks

b. Active attacks

a. Passive Attacks:- Passive attacks aims to learn or make use of information from the system, but does not affect the system resources. (Eavesdroping)

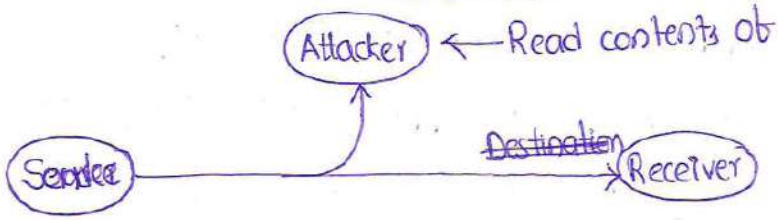
There are two types of passive attacks,

i) Release of message contents (Message Disclousure)

ii) Traffic Analysis

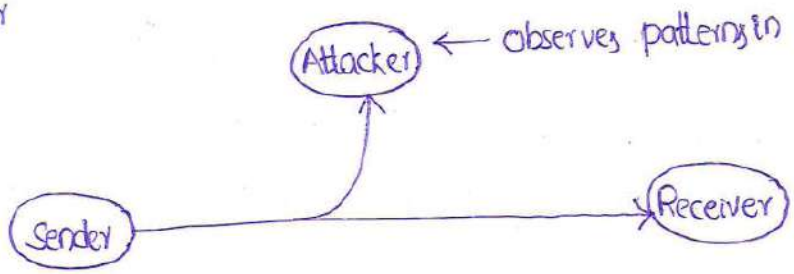
i) Release of message contents: In this attack, the attacker can read the contents of message during transmission.

Ex: Telephone line tapping



ii) Traffic Analysis:- Traffic analysis is the process of examining messages inorder to deduce information from patterns in communication.

Ex: Sniffer



* Passive attacks are very difficult to detect, because they do not involve any alteration of the data.
* Neither sender nor the receiver is aware that a party has read the messages or observed the traffic pattern.

It is feasible to prevent these attacks by means of encryption

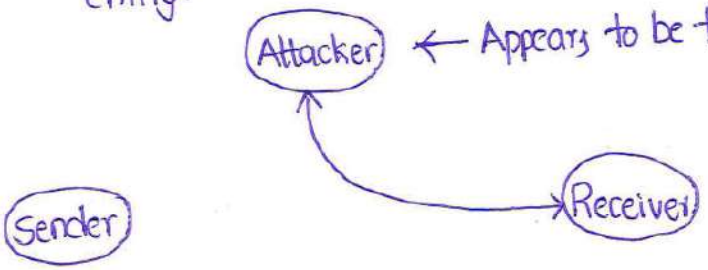
b. Active Attacks:-

Active attacks involve some modification of data stream or the creation of a false stream.

→ Active attacks are subdivided into four types:

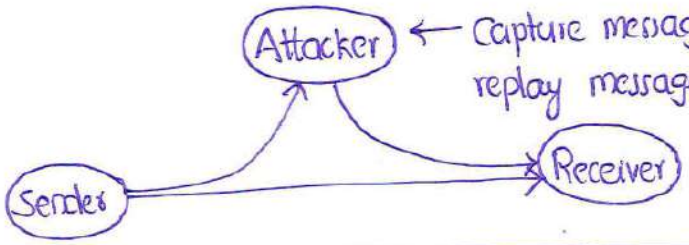
i) Masquerade
ii) Replay
iii) Modification of message
iv) Denial-of-service (DoS)

i) Masquerade: It takes place when one entity pretends to be a different entity.



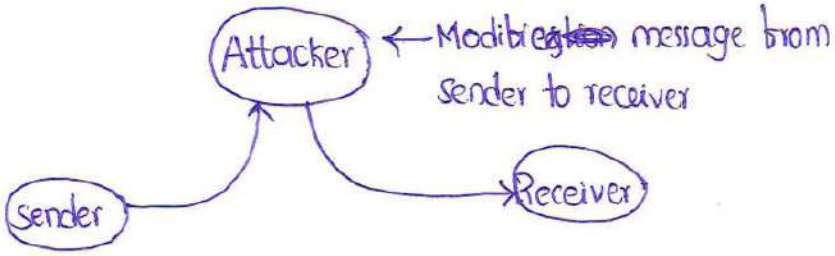
* Masquerade attacker gains access to the account of a legitimate user either by stealing the victim's account ID and password.

ii) Replay: Replay attack involves the passive capture of a data unit and its subsequent retransmission to produce an unauthorized affect.



One of the best techniques to prevent replay attacks is by using strong digital signatures with timestamps.

iii) Modification of Messages:- Modification of messages means that some portion of legitimate message is altered or reordered to produce an unauthorized object.



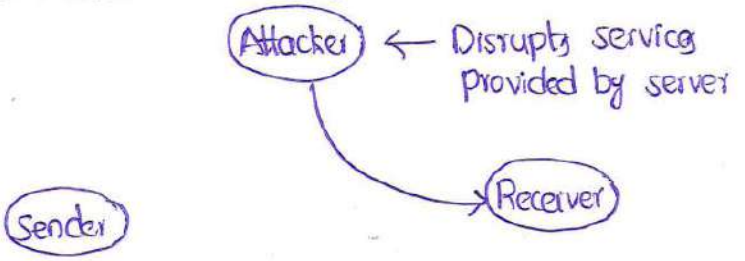
Ex: Original message: "Transfer 10,000 to account no: 123456"
Modified message: "Transfer 10,000 to account no: 456789"

The best method to prevent modification of messages attack is cryptographic hash functions.

iv) Denial-of-Service (DoS): Denial-of-service attack is an attempt to mak a machine or network resource unavailable to it's intended users.

* Categories of resources which can be attacked:

1. network bandwidth 2. System resources 3. Application Resources



DoS prevention techniques:

1. Seperate Client and Server Addresses
2. Non Global client addresses
3. Middle walls

It is difficult to prevent active attacks because of the wide variety of potential physical, software and network vulnerabilities

2. Security Mechanisms:—

Security mechanisms are used to implement security services.

→ Security mechanisms include:

a. Encipherment
b. Digital Signature
c. Access Control Mechanisms
d. Data Integrity Mechanisms
e. Authentication Exchange
f. Traffic Padding
g. Routing control
h. Notarisation.

a. Encipherment: It is a security mechanism that involves the transformation of data into some unreadable form.

* Encipherment ensures privacy (confidentiality) by keeping the ~~hidden~~ information hidden from any one for whom it is not intended.
* Decipherment is the reverse of encipherment.
* Encipherment is performed on plaintext (readable data) to produce ciphertext (unreadable data).
* Encipherment and Decipherment require the use of some secret information.

b. Digital Signature: Digital signature is a cryptographic value that is calculated from the data and a secret key.

* Digital signatures are used to validate the authenticity and integrity of a message, software or digital document.
* Digital signatures are based on public key cryptography.

c. Access Control Mechanisms: Access control mechanisms are a set of controls to restrict access to certain resources.

* There are two main types of access control:

1. physical: physical access control limits access to campuses, buildings, rooms and physical IT assets.
2. Logical: Logical access control limits access to connections to computer networks, system files and data.

* There are four categories of access control mechanisms:

1. Mandatory Based control:
2. Discretionary access control
3. Role-Based access control
4. Rule Based access control

* Access control mechanisms provide authentication and integrity.

d. Data Integrity Mechanisms: A variety of mechanisms used to assure the integrity of a data unit or stream of data units.

The following mechanisms are used to provide integrity,

1. Encipherment
2. Digital Signatures

e. Authentication Exchange: A mechanism intended to ensure the identity of an entity by means of information exchange

f. Traffic Padding: The insertion of bits into gaps in a data stream to prevent traffic analysis attack

g. Routing Protocol: Enables selection of particular physically secure routes for certain data and allows routing changes.

h. Notarisation: The use of a trusted third party to assure certain properties of a data exchange.

3. Security Services: Security service is a service which ensures adequate security of the systems or data transfers.

→ X.800 divides security services into 5 categories:

a. Authentication: Authentication is a process that ensures and confirms user's identity.

Ex: Consider a person, using online banking service. Both user and the bank should be assured in identities of each other.

b. Access Control: Access control is any mechanism of limiting access to a system or to physical or virtual resources.

→ Access control is a process by which users are granted access and certain privileges to systems, resources, or information.

Ex: In online banking, a user may be allowed to see his balance, but not allowed to make any transactions for some of his accounts.

c. Data Confidentiality: The protection of data from unauthorized users.

→ There are four types of Data confidentiality:

i. Connection confidentiality
ii. Connectionless confidentiality.
iii. Selective field confidentiality.
iv. Traffic-Flow confidentiality.

d. Data Integrity: Data integrity is the assurance that data received are exactly by an authorized entity.

→ The data contain no modification, no insertion, no deletion or no replay.

→ Data integration provides protection from active attacks.

e. Non Repudiation: Non repudiation refers to the ability to ensure that a party to a contract or a communication cannot deny authenticity of their signature.

→ It provides protection against denial by one of entinties in communication.

→ Non repudiation can be related to

Origin: proof that the message was sent by specified party.

Destination: proof that the message was received by specified party

f. Availability: Availability refers to the ability of a user to access information or resources in a specified location and in the correct format.

→ Availability addresses denial-of-service attacks.

② Model For Network Security:-

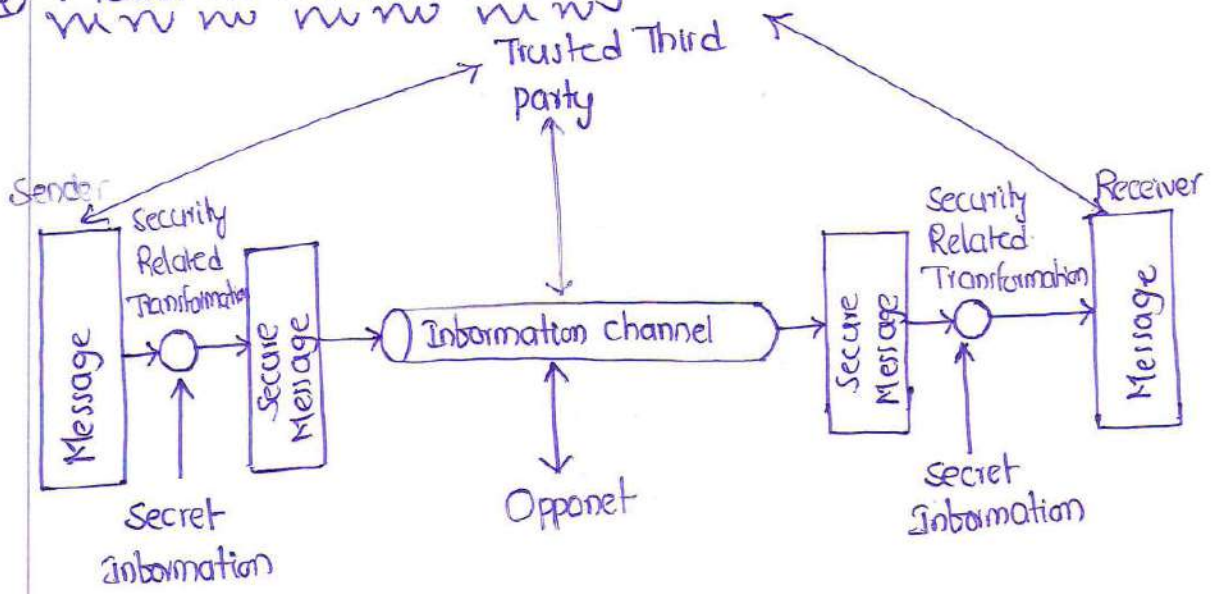


Fig: Model for Network Security

→ Message to be transferred from one party to another across some sort of internet.

→ The two parties must cooperate for the exchange to take place.

→ A logical information channel is established by defining a route through internet from source to destination.

→ Security aspects come in to play when it is necessary to protect the information transmission from an opponent.

→ All the techniques for providing security have two components:

① Security Related Transformation

② Secret Information

→ The General model for designing a particular security service:

① Design an algorithm for performing security-related transformation.

② Generate the secret information to be used with the algorithm.

③ Develop methods for distribution and sharing of secret information.

④ Specify a protocol to be used by the two principals and makes use of security algorithm and secret information.

③ Symmetric Cipher Model:—

→ Symmetric cipher model is also called "conventional / private-key / single key" ~~encryption~~ cipher.

→ In symmetric cipher model, sender and receiver share a common key.

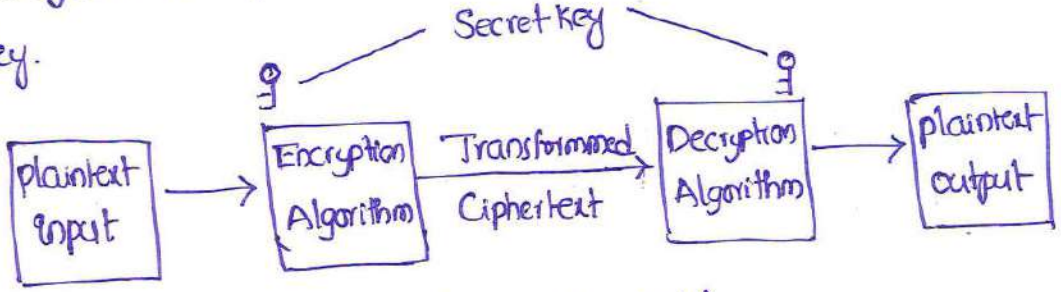


Fig: Symmetric cipher model

→ Symmetric cipher model contains five ingredients,

1. plaintext: The original message to be transmitted.

2. Encryption Algorithm: It performs various substitutions / transformations on plaintext.

3. Secret key: Secret key is a value independent of plaintext and the algorithm. The substitutions/transpositions depend on the key.

4. Cipher Text: The transformed message produced as output. It depends on plaintext and secret key.

5. Decryption Algorithm: It takes the ciphertext and secret key and produces the original plaintext.

→ Cryptography: The science and art of transforming messages to make them secure is called "Cryptography".

→ Encryption: The process of converting plaintext to ciphertext is called "Encryption".

→ Decryption: The process of converting ciphertext to plaintext is called "Decryption".

→ There are two basic requirements for encryption:

1. Strong encryption algorithm
2. Secret key must only known to sender/receiver.

→ Let X be plaintext, Y be ciphertext then

Encryption : $Y = E_K(X) = E(K, X)$
Decryption : $X = E_K(Y) = D(K, Y)$

→ Symmetric encryption can be done using:

1. Substitution Ciphers
2. Transposition Ciphers.

1) Substitution Ciphers:-

→ Substitution cipher is one in which the letters of plaintext are replaced by other letters or by symbols or by numbers.

→ Substitution cipher changes characters in the plaintext to produce ciphertext.

a) Caeser Cipher:-

→ Caeser cipher is the earliest known substitution cipher used by Julius caeser (Roman Emperor) to send messages to his army.

→ Caesar cipher involves replacing each letter of the alphabet with the letter standing 3 places further down the alphabet.

Ex: plaintext: qiscet
ciphertext: TLVFHW

*The alphabet is wrapped around 'z'

Encryption:

$$C = (P+3) \bmod 26$$

Decryption:

$$P = (C-3) \bmod 26$$

C - ciphertext
P - plaintext

Example: plaintext: s e c u r i t y
(18 4 2 20 17 8 19 24)
key + 3 3 3 3 3 3 3 3
(21 7 5 23 20 11 22 27) mod 26
↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
ciphertext: V H F X U L W B

→ Caeser cipher is the weakest cipher because key value is fixed.

b) Shibt Cipher:-

→ Shibt Cipher substitutes each letter in the plaintext with k^{th} letter following in the alphabet. Wrapping around when we get z. (Shibt Rig

→ Decryption is the reverse process of encryption (Shibt Left)

Encryption:	$C = (P+K) \bmod 26$
Decryption:	$P = (C-K) \bmod 26$

$K = 0, 1, 2 \cdots 25$

Ex: $k = 7$

		18	4	2	20	17	8	19	24
plaintext:		S	e	c	u	r	i	t	y
key	+	7	7	7	7	7	7	7	7
		25	11	9	27	24	15	26	31
									mod 26
Ciphertext:		Z	L	J	C	Y	P	A	F

Demerits:

* Simple structure usage
* Provide minimum security for information
* key space is very small (1 to 25)
* It can be breaked in 26 posibilities.

c) Monoalphabetic Substitution Cipher:

→ Monoalphabetic substitution ciphers changes a character in the plaintext to the same character in the ciphertext for any occurance.

→ Rather than shifting, this cipher shuffle the letters in the plaintext with an arbitrary substitution.

→ Key is the random permutation applied on characters in alphabet.

→ There can be 26! possible keys.

Example: key

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

→ Let $\Pi(p)$ represent permutation for a plaintext P.

Encryption:	$C = \Pi(P)$
Decryption:	$P = \Pi^{-1}(C)$

Π^{-1} ← Inverse permutation

Ex: plaintext: s e c u r i t y } Encryption (permutation)
Ciphertext: M S Q U N E J O
plaintext: s e c u r i t y } Decryption (Inverse permutation).

Demerits:

* The attacker can make guesses by observing the relative frequency of letters, digrams and trigrams in the text.
* Easy to break in general.
* It can be breaked in 26! possibilities.

d) Polyalphabetic Substitution Ciphers:-

→ Polyalphabetic substitution ciphers replace a character in the plaintext with different character in ciphertext for each occurance.

i) Vigenere Cipher:

→ Vigenere cipher was invented by Frenchman, Blaise de Vigenere.

→ It is a simple polyalphabetic cipher.

→ Key: The vigener cipher choose a sequence of keys, represented by a string.

* The key letters are applied to successive plaintext characters and when end of key is reached, the key start over.
* The length of the key is called "period" of the cipher.

→ Given m, a positive integer $P=C=(Z_{26})^m$ and

$K=(k_1, k_2 \cdots k_m)$ a key

Encryption: $E_K(P_1, P_2, \cdots P_m) = (P_1+k_1, P_2+k_2 \cdots P_m+k_m) \pmod{26}$

Decryption: $D_K(C_1, C_2, \cdots C_m) = (C_1-k_1, C_2-k_2 \cdots C_m-k_m) \pmod{26}$.

Example: plaintext: n e t w o r k s e c u r i t y

key = vig key (+) v i g k e y v i g k e y v i g

ciphertext: (54 12 25 32 18 41 31 26 10 12 24 41 29 27 30) mod

C M Z G S P F A K M Y P D B E

→ Strength of Vigenere cipher:

The strength of vigenere cipher is that there are multiple ciphertext letters for each plaintext letter, one for each unique letter of keyword.

ii) One time pad:

→ A one-time pad is the key which is created by generating a string of characters and numbers that will be atleast as long as the plain-text.

→ The one-time pad consists of random characters or numbers

→ The OTP should only be used once and destroyed by sender and receiver after it's use.

Encryption:	$C = P + OTP$
Decryption:	$P = C - OTP$

Example: plaintext: s e c u r i t y

OTP(key): x a w m p v t c

$(41\ 4\ 44\ 32\ 32\ 29\ 38\ 26) \bmod 26$

Ciphertext: P E S G G D M A

→ OTP is Unbreakable:

* Key is based on randomness. It has the advantage that theoritically no way to break the code by analyzing succession of message
* Each encryption is unique and bears no relation to next encryption so that pattern can be detected
* The key is destroyed after use by the sender and receiver.

Demerits:

* Distribution of the key is a challenge.
* Adding plaintext and key manually is a time consuming task.

(iii) Vernam Cipher:

→ Vernam Cipher is an variant of one-time-pad

→ Vernam cipher was developed by Gilbert Vernam in 1918.

→ The plaintext is represented as a sequence of 0's and 1's

→ The key is also a random sequence of 0's and 1's of same length as the plaintext.

Encryption:	$C_i = P_i \oplus K_i$
Decryption:	$P_i = C_i \oplus K_i$

Example: plaintext : 0 1 0 1 1 1 0 0

key : $\oplus$ 1 0 0 0 1 1 0 1

Ciphertext: 1 1 0 1 0 0 0 1

→ Binary random key sequence have two fundamental properties:

① Unpredictability: Independent of the no of bits of a sequence observed, the probability of guessing the next bit is not better than $\frac{1}{2}$.

② Balanced: The number of 1's and 0's should be equal.

e) PlayFair Cipher:-

→ Play Fair cipher was invented by Wheatstone in 1854, but it was promoted by Lord Playfair.

→ It is a digraph substitution cipher. (Pairs of letters to be encrypted).

→ Key:

* Playfair cipher uses a 5x5 matrix
* The key should not have repeating letters.

→ Creating the 5x5 matrix:

1. Place I and J in the same grid of matrix
2. Insert all the characters in ~~plaintext~~ keyword in to matrix
3. Insert rest of characters in alphabet in to matrix.

Example: ~~key = "PLAY"~~

key = SECURITY

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

→ Preparing the plaintext:

* Split the plaintext into pairs (digraphs)
* Seperate all duplicate letters by inserting letter 'X'.
* If there is an odd no. of letters in plaintext, add 'X' at the end of message.
* Ignore all spaces, and punctuation marks.

Example:

Duplicate characters.

plaintext: SE CRET ME(SS)AGE

Insert X between two s's.

SE CR ET ME SX SA GE

Encryption: Lookout for the characters in the digraph in 'key' matrix.

1. If the characters in the digraph are in the same row
   * Replace the characters with the characters right to it
   * Upon reaching the end of row, wrap around.
2. If the characters in the digraph are in the same column
   * Replace the characters with the characters down to it
   * Upon reaching the end of column, wrap around.
3. If the characters in the digraph form a rectangle
   * Replace the characters with the characters on the opposite corners of rectangle

Ex: plaintext: SE CR ET ME SX SA GE

digraph SE: S, E are in same row.
Replace them with EC

digraph CR: C, R are in same row
Replace them with US

digraph ET: E, T are in same column
Replace them with TF

digraph ME: M, E are in same colum
Replace them with VT

digraph SX: S, X forms a rectangle
Replace them with U Q

digraph SA: S, A forms a rectangle
Replace them with UI

digraph GE: G, E forms a rectangle
Replace them with FC

Ciphertext: EC US TF VT UQ UI FC.

Decryption:

① If the characters in the digraph are in same row,
* Replace them with the characters in the left

② If the characters in the digraph are in same column,
* Replace them with the characters in the above

③ If the characters in the digraph are in a rectangle,
* Replace them with the characters in the opposite corners.

④ Remove any extra 'X' in the decrypted text to reveal final plaintext

f) Hill Cipher:-

→ Hill Cipher is invented by Lester S Hill in 1929.

→ It is based on linear algebra.

→ It is a polygraphic substitution cipher. It encrypts characters in groups.

→ Hill cipher uses matrices to encrypt and decrypt.

→ It uses modular arithmetic (mod 26).

→ Modular Inverse:

Modular Inverse for mod m : $(a\, a^{-1}) \bmod m = 1$.

For modular inverses, a and m must not have any prime factors in common.

Ex: Modular inverse of 3 is

$(3 \cdot 9) \bmod 26 = 27 \bmod 26 = 1$.

So, 3 is modular inverse of 9. and vice versa.

a	1	3	5	7	9	11	15	17	21	23	25
a^{-1}	1	9	21	15	3	19	7	23	5	17	25

Key:

* Hill cipher uses a matrix as key for encryption and decryption.
* It uses one matrix for encryption and one matrix for decryption.
* The matrix must be an invertible matrix.
* Decryption matrix must be modular inverse of encryption matrix in (mod 26).

Ex: Encryption matrix : $\begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix}$

* The matrix should be a square matrix of size nxn.

→ Modular inverse of a matrix:

* Calculate determinant of encryption matrix.
* Make sure that the determinant has a modular inverse of mod 26.
* Calculate the adjoint of encryption matrix.
* Multiply adjoint matrix by modular inverse of determinant.
* Resultant matrix is the decryption matrix.

Example:

Encryption matrix = $A = \begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix}$

$\det A = (2 \times 4) - (1 \times 3) = 8 - 3 = 5$

Modular inverse of 5 mod 26 = 21

Decryption matrix = $B = 21 \begin{bmatrix} 4 & -1 \\ -3 & 2 \end{bmatrix} = \begin{bmatrix} 84 & -21 \\ -63 & 42 \end{bmatrix}$

Apply mod 26.

$B = \begin{bmatrix} 84 & -21 \\ -63 & 42 \end{bmatrix} \bmod 26 = \begin{bmatrix} 6 & -21 \\ -11 & 16 \end{bmatrix}$

Convert negative numbers to positive numbers by adding 26 to the negative number.

$B = \begin{bmatrix} 6 & -21+26 \\ -11+26 & 16 \end{bmatrix} = \begin{bmatrix} 6 & 5 \\ 15 & 16 \end{bmatrix}$

Decryption matrix = $B = \begin{bmatrix} 6 & 5 \\ 15 & 16 \end{bmatrix}$

Encryption:-

1) Assign each letter in alphabet a number from 0 to 25.

2) Change message into nx1 letter vectors.

3) change each vector into nx1 numeric vectors

4) Multiply each numeric vector by a nxn encryption matrix.

5) Convert product vectors to letters.

Example: Encryption matrix = $A = \begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix}_{2\times 2}$

plaintext: S E C U R I T Y.

Change plaintext to 2x1 vectors and assign numbers.

$$\begin{bmatrix} S \\ E \end{bmatrix} = \begin{bmatrix} 18 \\ 4 \end{bmatrix}$$

$$\begin{bmatrix} C \\ U \end{bmatrix} = \begin{bmatrix} 2 \\ 20 \end{bmatrix}$$

$$\begin{bmatrix} R \\ I \end{bmatrix} = \begin{bmatrix} 17 \\ 8 \end{bmatrix}$$

$$\begin{bmatrix} T \\ Y \end{bmatrix} = \begin{bmatrix} 19 \\ 24 \end{bmatrix}$$

Multiply each vector with encryption matrix $A = \begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix}$

$$\begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 18 \\ 4 \end{bmatrix} = \begin{bmatrix} 36+4 \\ 54+16 \end{bmatrix} = \begin{bmatrix} 40 \\ 70 \end{bmatrix} \bmod 26 = \begin{bmatrix} 14 \\ 18 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 2 \\ 20 \end{bmatrix} = \begin{bmatrix} 20+4 \\ 6+80 \end{bmatrix} = \begin{bmatrix} 24 \\ 86 \end{bmatrix} \bmod 26 = \begin{bmatrix} 24 \\ 8 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 17 \\ 8 \end{bmatrix} = \begin{bmatrix} 34+8 \\ 51+32 \end{bmatrix} = \begin{bmatrix} 42 \\ 83 \end{bmatrix} \bmod 26 = \begin{bmatrix} 16 \\ 05 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 19 \\ 24 \end{bmatrix} = \begin{bmatrix} 38+24 \\ 57+96 \end{bmatrix} = \begin{bmatrix} 62 \\ 153 \end{bmatrix} \bmod 26 = \begin{bmatrix} 10 \\ 23 \end{bmatrix}$$

Convert numeric vectors to letter vectors.

$$\begin{bmatrix} 14 \\ 18 \end{bmatrix} = \begin{bmatrix} O \\ S \end{bmatrix}$$

$$\begin{bmatrix} 24 \\ 08 \end{bmatrix} = \begin{bmatrix} Y \\ I \end{bmatrix}$$

$$\begin{bmatrix} 16 \\ 05 \end{bmatrix} = \begin{bmatrix} Q \\ F \end{bmatrix}$$

$$\begin{bmatrix} 10 \\ 23 \end{bmatrix} = \begin{bmatrix} K \\ X \end{bmatrix}$$

Ciphertext: OSYIQFKX

Decryption:

1) Change ciphertext to nx1 letter vectors.
2) change each vector to nx1 numeric vectors.
3) Multiply each numeric vector by decryption matrix.
4) Convert numeric vectors to letters to get the plaintext.

Example: Decryption matrix $B = \begin{bmatrix} 6 & 5 \\ 15 & 16 \end{bmatrix}$

Ciphertext: OS YI QF KX

Convert ciphertext to 2x1 vectors and assign numbers to letters.

$$\begin{bmatrix} O \\ S \end{bmatrix} = \begin{bmatrix} 14 \\ 18 \end{bmatrix}$$

$$\begin{bmatrix} Y \\ I \end{bmatrix} = \begin{bmatrix} 24 \\ 8 \end{bmatrix}$$

$$\begin{bmatrix} Q \\ F \end{bmatrix} = \begin{bmatrix} 16 \\ 5 \end{bmatrix}$$

$$\begin{bmatrix} K \\ X \end{bmatrix} = \begin{bmatrix} 10 \\ 23 \end{bmatrix}$$

Multiply letter vectors with Decryption matrix $B = \begin{bmatrix} 6 & 5 \\ 15 & 16 \end{bmatrix}$

① $\begin{bmatrix} 6 & 5 \\ 15 & 16 \end{bmatrix} \begin{bmatrix} 14 \\ 18 \end{bmatrix} = \begin{bmatrix} 84+90 \\ 210+288 \end{bmatrix} = \begin{bmatrix} 174 \\ 498 \end{bmatrix} \bmod 26 = \begin{bmatrix} 18 \\ 4 \end{bmatrix}$

② $\begin{bmatrix} 6 & 5 \\ 15 & 16 \end{bmatrix} \begin{bmatrix} 24 \\ 8 \end{bmatrix} = \begin{bmatrix} 144+40 \\ 360+128 \end{bmatrix} = \begin{bmatrix} 184 \\ 488 \end{bmatrix} \bmod 26 = \begin{bmatrix} 2 \\ 20 \end{bmatrix}$

③ $\begin{bmatrix} 6 & 5 \\ 15 & 16 \end{bmatrix} \begin{bmatrix} 16 \\ 5 \end{bmatrix} = \begin{bmatrix} 96+25 \\ 240+80 \end{bmatrix} = \begin{bmatrix} 121 \\ 320 \end{bmatrix} \bmod 26 = \begin{bmatrix} 17 \\ 8 \end{bmatrix}$

④ $\begin{bmatrix} 6 & 5 \\ 15 & 16 \end{bmatrix} \begin{bmatrix} 10 \\ 23 \end{bmatrix} = \begin{bmatrix} 115 + 60 \\ 150 + 368 \end{bmatrix} = \begin{bmatrix} 175 \\ 518 \end{bmatrix} \bmod 26 = \begin{bmatrix} 19 \\ 24 \end{bmatrix}$

convert numeric vectors to letter vectors.

$$\begin{bmatrix} 18 \\ 4 \end{bmatrix} = \begin{bmatrix} S \\ E \end{bmatrix}$$

$$\begin{bmatrix} 2 \\ 20 \end{bmatrix} = \begin{bmatrix} C \\ U \end{bmatrix}$$

$$\begin{bmatrix} 17 \\ 8 \end{bmatrix} = \begin{bmatrix} R \\ I \end{bmatrix}$$

$$\begin{bmatrix} 19 \\ 24 \end{bmatrix} = \begin{bmatrix} T \\ Y \end{bmatrix}$$

plaintext: SECURITY

Advantages:

* It completely hides single letter frequency.
* It is strong against ciphertext only attack
* By using larger size matrix, more frequency information hiding is possible

Disadvantages:

* It is difficult to calculate modular inverse of a matrix.
* Easily broken with known plaintext attack.

(5) Transposition Techniques:

→ The transposition cipher rearranges the characters in the plaintext to form the ciphertext.

→ The letters in the plaintext are not changed.

i) Rail Fence Technique:

→ The Rail Fence cipher is composed by writing the plaintext into rows proceeding down, then across and reading the ciphertext across and then down.

Ex: plaintext: cryptography and network security

c y t g a h a d e w r s c r t
r p o r p y n n t o k e u i y } depth = 2

ciphertext: cytgahadewrscrtrporpynntokeuiy

Disadvantage:

* It is not particularly secure, since there are a limited number of usable keys, especially for short messages.

ii) Columnar Transposition:

→ In columnar transposition cipher, the plaintext is written out in rows of a fixed length. The plaintext is then read out column by column to make ciphertext.

→ The number of columns and the order in which they are chosen is defined by a keyword.

→ The order is choosen by the alphabetical order of the letters in the keyword.

Ex: keyword: C I P H E R
↓ ↓ ↓ ↓ ↓ ↓
permutation: 1 4 5 3 2 6

Example: plaintext: SIMPLE TRANSPOSITION

C	I	P	H	E	R
1	4	5	3	2	6
S	I	M	P	L	E
T	R	A	N	S	P
O	S	I	T	I	O
N	X	X	X	X	X

Fill the remain row entries with a filler like X.

Readout in columns.

Ciphertext: STON LSIX PNTX IRSX MAIX EPOX

→ Decryption:

* write out the keyword
* Divide the ciphertext into columns of equal letters.
* Put the first column of letters under the letter in the keyword that comes alphabetically first.
* Put the second column of letters under the letter in the keyword that comes second, etc.
* Readout the data in rows to make the plaintext.

??) Double Columnar Transposition:

→ Double columnar Transposition is used to make more security for the transposition cipher.

Example: consider the result of above columnar transposition

C	I	P	H	E	R
1	4	5	3	2	6
S	T	O	N	L	S
I	X	P	N	T	X
I	R	S	X	M	A
I	X	E	P	O	X

Ciphertext:

SIIITXRXOPSE
NNXP LTMOSXAX

(6) Cyber Threats & Defense:-

Cyberthreat: The possibility of a malicious attempt to damage or disrupt a computer network or system.

→ In this section, the following cyberthreats to be discussed,

a) Phishing
b) SQL Injection
c) Web Based Attacks

a) Phishing Attacks:

→ Phishing is the fraudulent process of attempting to aquire sensitive information suchas usernames, passwords, credit card details, bank account details and ebay etc. by masquerading as a trustworthy entity.

→ Phishing is a form of social engineering attack.

→ phishing has three components:

1. Mail sender: Sends large volumes of fraudulent emails.
2. Collector: Collects sensitive information from users.
3. Cashier: Uses the collected information to encash.

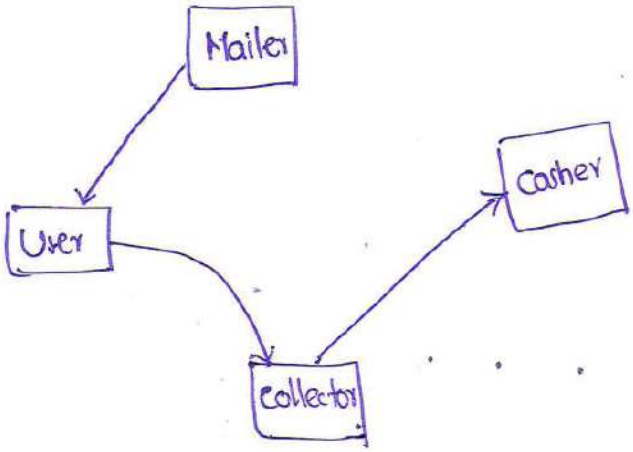


Fig: Phishing Information flow.

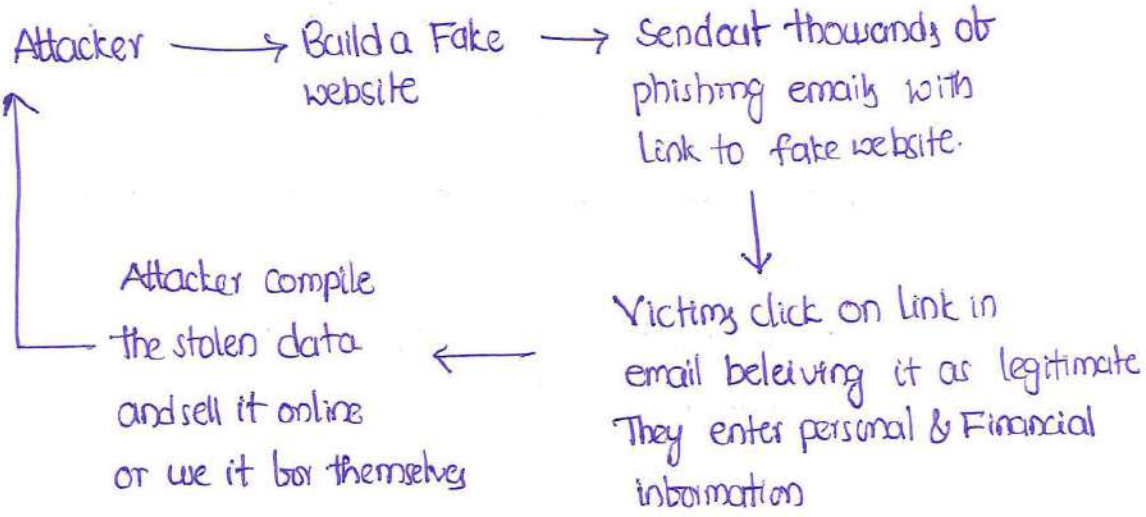
Phishing Forms:

* Creating Fake URLs and send it
* Mispelled URLs

  Ex: www.micosoft.com
* creating anchor text

  Ex: Link text
* Fake SSL lock
* Getting valid certificates to illegal sites.
* URL Manipulation using Javascript.

Working of Phishing:



Defenses for Phishing Attack:

* Never click on hyperlinks. never cut and paste the link, type the url to go to website in search engine
* call the company directly to confirm whether the website is valid
* Don't reply to email or pop-up messages that ask for personal or financial information
* Don't email personal and Financial information
* Be cautious in opening attachments.

b) SQL Injection Attacks:

→ An SQL injection attack involves placing SQL statements in the user input.

→ SQL Injection is a code injection technique that exploits a security vulnerability occuring in the database layer of an applicatio

Example:

productsearch: `blah' OR 'x'='x`

* This input is put directly into SQL statement within the web application:

```
$query = " SELECT productinbo FROM prodtable WHERE
          prodname = ' " . $_post['pod_search'] . " ' ";
```

* Creates the Following SQL:

```
SELECT prodinbo FROM prodtable WHERE prodname = 'blah' OR
'x' = 'x';
```

The attacker has now successfully caused the entire database to be returned.

→ Other SQL Injection Possibilities:

* Add new data to the database
* Modify data currently in the database
* Gain access to other user's system capabilities by obtaining their password
* Delete the entire database

Ex: `blah'; DROP TABLE prodinbo; --`

Defenses for SQL Injection Attacks:

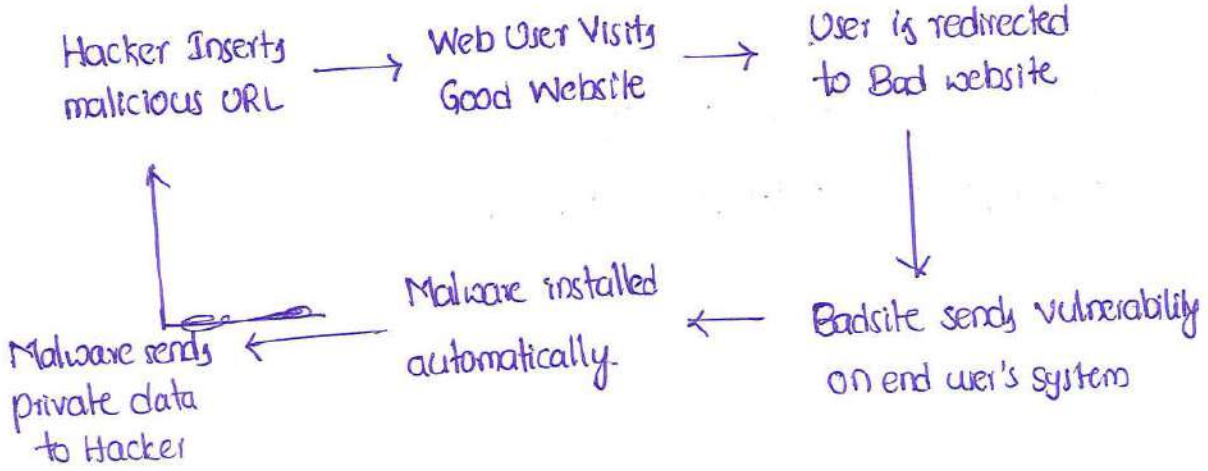
* Use bound variables with prepared statements.
* Use provided functions for escaping strings.
* check syntax of input for validity.
* Provide length limits on input.
* Scan query string for undesirable word combinations that indicate SQL statements.
* Limit database permissions and segregate users.
* Configure database error reporting

c) Web Based Attacks:-

→ Web based attack is any threat that uses the WWW (World Wide Web) to facilitate cybercrime.

→ Web based attacks use multiple types of malware and fraud, all of which utilize HTTP and HTTPS protocols, email or IM (Instant Messenger).

Anatomy of Web Based Attacks:



1) Attacker breaks into legitimate website and post malware.

2) Attacking end-user machines.

3) Leveraging end user machines for malicious activity.

→ Clickjacking: The click of link executes the attacker's code, often leading the person to a malicious website.

→ Types of Social Engineering Attacks:

1. Fake Codec: In this user is prompted to instal missing codec but the codec is actually a malware (Trojan Horse).

2. Malicious Peer-to-Peer files: The Malware authors bind the content into popular applications

3. Malicious Advertisements: Malware Authors advertise their fake codecs to unsuspecting users.

4. Fake scanner: Create a website or product that misrepresents the truth.

Defenses for Web Based Attacks:

* Update and patch the software
* Avoid the things that seems too good
* Use safe search functionality in browsers.
* Adopt strong ~~passs~~ password policy.

7) Bubber Overflow:-

→ Bubber overflow occurs when a program or process tries to stores more data in a bubber than it was intended to hold.

→ Bubbers are created to contain binite amount of data, the extra information can overflow into adjacent bubbers, corrupting or overwriting the valid data held in them.

→ Bubber overflow is an attack on integrity.

→ A bubber for a program or process is allocated either in stack area or heap area of memory.

→ The bubberoverflow can occur in stack or heap.

Ex:
```
int i;
void function(void)
{
  char bubber[256];      // create a bubber
  for(i=0;i<512;i++)     // iterate 512 times
      bubber[i] = 'A';   // copy the letter 'A'
}
```

The above program declares a bubber that is 256 bytes long. However the program attempts to bill with 512 bytes of letter 'A'.

→ Some ~~wes~~ unsafe functions in C that cause bubberoverflow are:

strcpy()
strcat()
gets()
scanf()
printf()

Defense against Buffer Overflow:

* Use only the good form of printf(). Never use printf(buffer) for any function
* Review loop bounds
* Avoid unsafe C functions.
* Inserts bound checking code.
* Avoid unsafe programming languages. ~~like Java, C++~~.
* static source code analysis
* Mark stack as non-execute.
* Use safe languages like java and C++.
* The safe language should check bounds on buffer.

(8) Format String Attack:-

→ Format string specifies a method of rendering an arbitrary number of varied datatype parameters into a string.

→ The Format string is printed on the standard output stream.

Eg: printf("The magic number is %d\n", 1911);

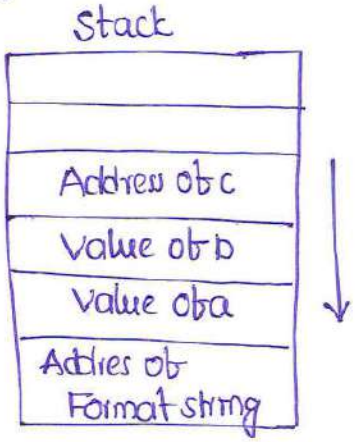
The text to be printed is "the magic number is" followed by %d format specifies which is replaced with 1911 in output.

→ Format Parameters:

%d : decimal
%u : unsigned decimal
%x : hexadecimal
%s : string
%n : no. of bytes written so far.

→ Stack and it's role at format string:

The behavior of format function is controlled by the format string. The function retrieves the parameters requested by the format string from the stack.



Ex: printf("a has value %d,
b has value %d,
c is at address %08x\n",
a, b, &c);

→ If the printf() detect a miss-match

* The printf() fetches the arguments from stack. If the format string needs 3 arguments, it will fetch 3 data items from the stack.
* In the miss-match case, the printf() will fetch some data that do not belong to this function call.

→ Attacks on Format String:

1. Crashing the program:

printf("%s %s %s %s %s %s %s");

For each %s, printf() will fetch a number from stack, treat this number as an address and printout the memory contents pointed by this address as a string, untill a NULL character is encountered.

The memory pointed by this number might not exist, and the program will crash.

2. Viewing the stack:

printf("%08x %08x %08x %08x %08x\n");

It retrieves five parameters from the stack and display them as 8-digit padded hexadecimal numbers.

3. Viewing memory at any location:

→ If we use printf("%s") without specifying a memory address, the target address will be obtained from the stack. The function maintain an initial stack pointer, so it knows the location of parameters in the stack.

4. Writing an integer to any location in the process memory.

→ %n : The no of characters written so far is stored into the integer indicated by corresponding argument.

Ex: int i;
printf("12345 %n", &i);

→ It causes printf() to write 5 in to variable i.

Defenses for Format String Attacks:

* Always specify a format string as part of program, not as input.
* Make the format string a constant.
* Address Randomization: Randomization makes it difficult for the attacker's to find out what address they want to read/write.

→ The Format string attacks can be done using

printf()
sprintf()
fprintf()
snprintf()

⑨ Session Hijacking:

Session Hijacking is the act of taking control of a user session after successfully obtaining of an authenticate session id.

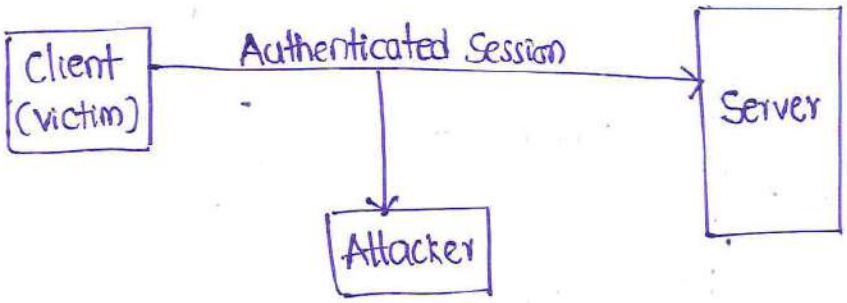
→ Session Hijacking involves an attack using captured session id to grab control of legitimate user.

→ session Hijacking takes place at transport layer of OSI.

a) TCP Session Hijacking:

→ The attacker takes control of a TCP Session between two hosts.

→ It can be hijacked after hosts have authenticated successfully.



→ To Hijack the session in the TCP, the Hijacker uses the following techniques.

i. IP Spoofing: (Source Routed Packets) | IP Spoofing is a technique used to gain unauthorized access to computers, whereby the intruder sends message to a computer with an IP address indicating the message is coming from a trusted host.

ii. Blind Hijacking: In blind hijacking the hijacker injects malicious data into intercepted communication in TCP session.

*It is called "blind" because the hijacker can send data or commands, but cannot see the response.

iii. Man-In-The-Middle Attack (Packet Snibbing):

This technique involves using a packet snibber that intercepts communication between client and server.

b) UDP Session Hijacking:

→ Since UDP doesnot use packet sequencing and synchronizing.
→ It is easier than TCP to hijack UDP session.
→ The Hijacker has simply to borge a server reply to a client UDP request before the server can respond.
→ If snibbing is used than it will be easier to control traffic generatin from the side of the server and thus restricting server's reply to the client.

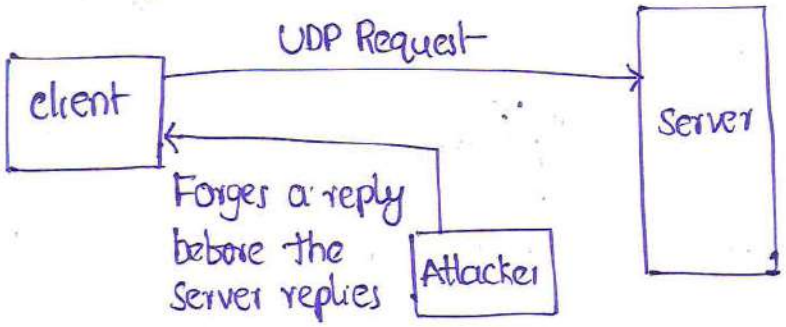


Fig: UDP Session Hijacking.

Defenses for Session Hijacking:

1. Encryption
2. Connections
3. Anti-Virus Software
4. Employee education

10) ARP Attacks:

→ ARP (Address Resolution Protocol) maps IP Address to MAC Address.

Ex: IP-MAC Binding

IP	MAC	TYPE
10.0.0.2	00:00:00:00:00:02	dynamic

→ ARP Spoofing is a technique used to attack an Ethernet network which may allow attacker to sniff data frames on a LAN or stop the traffic.

→ The ARP Spoofing principle is to send fake or spoofed, ARP messages to an Ethernet LAN. These frames contain false MAC Addresses, confusing network devices. As a result the frames can be mistakenly sent to other host or unreachable host.

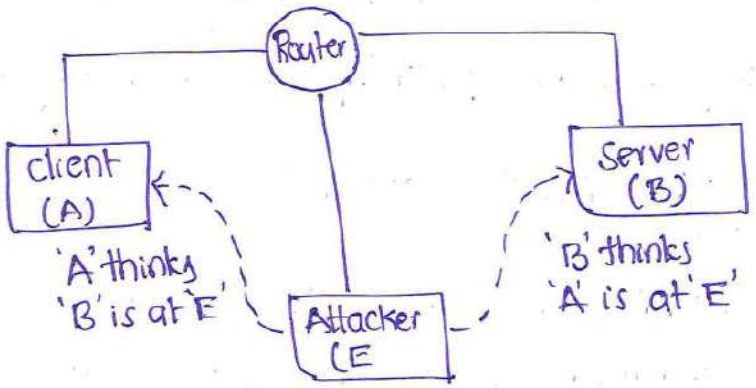


Fig: ARP Spoofing.

Defenses against ARP Spoofing:

1. Packet filters
2. Static Mapping of IP address to MAC address
3. ARP spoofing detection software
4. Cryptographic Network Protocols: Transport Layer Security (TLS)
   Secure Shell (SSH)
   HTTP Secure (HTTPS).

(11) Route Table Modification:

→ Route Table is a data table stored in a router that lists the routes to particular network destinations.

→ The Router moves the packets by looking into the routing table.

→ The Routing table is formed by exchanging routing information betwe routers.

→ Routing table modification means the unwanted or malicious change in route table of the router. This is done by editing the routing information update packets which are advertised by routers.

Defense for Route Table Modification:

* Routing protocol authentication and verification
* passive interfaces are used to stop sending updates on interfaces.
* Route Filtering to reduce possibility of false routing information.

(12) Man-In-The-Middle Attack: (MITM)

→ Man-In-The-Middle attack is an attack in which attacker is able to read, insert and modify messages between two parties.

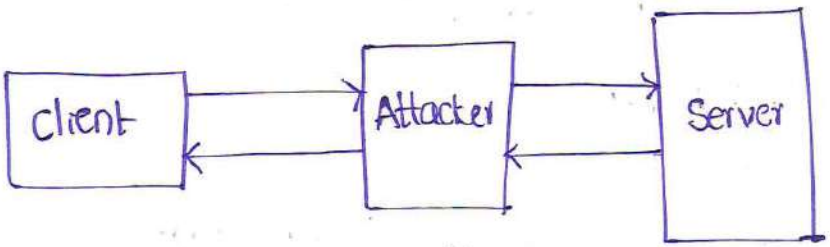


Fig: Man-In-The-Middle Attack.

→ The attacker able to observe and intercept messages going between two victims.

→ The MITM attack may include one or more of

1. Eavesdropping.
2. chosen ciphertext attack
3. Substitution attack
4. Replay attacks.
5. Denial-of-service attack

→ MITM used to refer active manipulation of messages. rather than passively eavesdropping.

Defenses against MITM attack:

1. public keys
2. Stronger Mutual Authentication
3. Secret keys
4. Passwords
5. Voice recognition and biometrics.