# Unit 4 - Introduction to Network Security



## > Importance and need for security

Security is imp to protect data and systems from being accessed or misused by unauthorized people. It helps to keep info private, accurate and available whenever we need it.

→ Protects data → maintains trust → keeps services running
→ stops unauthorized access → fights cyberattacks eg: banking / exam portals

SECURITY PRINCIPLES / GOALS

1) confidentiality - keeping data hidden from unauthorized access. Organizations must guard against hackers / threats that may reveal sensitive info.
Eg - While sending bank details online, it should be encrypted, so no one else can read it during transmission.

2) Integrity - changes need to be done by authorized entities & through authorized mechanism only. If the data is changed by hacker or even by system error like power failure, integrity is lost. So, integrity also protects against accidental changes.
Eg - If in a bank, someone deposits ₹1000, the system must update correctly, else some hacker changes to ₹10000, the data integrity is violated.

3) Availability - data should be available to an authorized person when it is needed. Unavailability can happen due to attacks (like DDos), system crashes or power issues. Even if data is safe, it's useless if user cant access it.
Eg. If Banking app is down, customers cant transfer.

SECURITY ATTACKS
- → Threat to confidentiality — Snooping, Traffic analysis
- → Threat to integrity — Modification, Masquerading, Replaying, Repudiation
- → Threat to availability — Denial of service

## > Network Attacks

Passive Attacks - The attacker only monitors or reads the data being transferred, but does not make any changes to it.
GOAL - To steal or collect information without being detected.

1) Eavesdropping / Snooping : The attacker listens to private conversation or reads messages sent over the network.
Eg. Hacker reads your emails secretly, that was sent on unprotected WiFi

2) Traffic analysis: The attacker checks who is talking to whom, how often and how much data is sent - even without reading the content
Eg. An attacker finds out which website you are visiting frequently, even if they can't see the content.

Passive attacks do not affect the system performance & are hard to detect

Active attacks - The attacker tries to change, damage or interrupt the data or services.

GOAL - To modify data, disrupt communication or gain unauthorized access.

1) Modification of messages : Changing data in transit. The attacker modifies the info to make it beneficial to herself.
Eg - A hacker changes the amount in an oline bank transfer from £1000 to £10,000

2) Masquerade attack - Attacker pretends to be someone else (like an admin) ie authentic user.
Eg. Hacker logs in using someone else's identity to gain access to private systems.

3) Replaying : When an attacker captures a valid message and resends (replays) it to trick the system.
Duplicates valid transactions & gain an advantage like getting paid twice.

4) Repudiation Attack - This is performed by one of the two parties in the communication : the sender/receiver. One party in communication denies taking an action. Not attacked by any outsider.
Eg. A person buys a product online & makes the payment, but seller denies receiving the payment & demands it again.

5) Denial of service (DoS) Attack : Making a service (like a website) unavailable by overloading it with fake requests.
Eg. Shopping website crashes because a hacker floods it with fake users.

Active attacks affect system integrity & availability and are usually easier to detect.

> Network Security Threats

Threats - Any activity or event that violates security goals ie confidentiality, integrity or availability of info/services over a network.

Unauthorized Access - Occurs when a person or system accesses data without any permission. Hackers or malicious users try to bypass authentication (like passwords or biometric locks) to enter into private systems.
Eg. Hacker logs into company's server using stolen login credentials & downloads private business files.

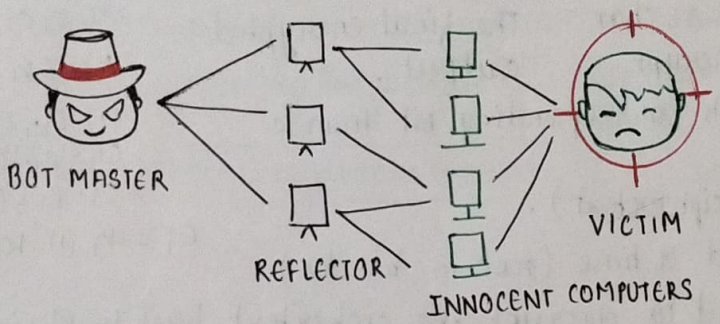
Distributed Denial of Service (DDoS) Attacks - Malicious attempt to disrupt the normal traffic of targeted server or network by overwhelming it with a flood of internet traffic from multiple sources.
Typical Targets of DDoS are : Routers, Links, Firewalls, Internet shopping sites, etc

Types of DDoS -

1) Volume Based Attacks : These attacks try to consume all the available bandwidth between the target and the internet. They use huge volume of traffic to flood the network. Eg. UDP flood, ICMP flood, DNS Amplifier

2) Protocol Attacks - These attacks consume server resources like memory by abusing protocol features. Sends specially crafted packets that make the targets system work harder or get stuck. Eg. SYN flood, Ping of death, Smurf Attack

3) Application Layer Attacks - These attacks target the application or web server directly, mimicking legitimate user requests but in very large numbers. They overload specific functions like login pages or search features. Eg. HTTP Get/Post flood, WordPress XML-RPC Attack, Slowloris, etc.




In 2016, a major DDoS attack using Mirai botnet took down big websites like Twitter, Netflix & Reddit by overwhelming a DNS provider with traffic from millions of IoT devices.

Man-in-the-middle Attack (MITM)

MITM is an eavesdropping situation in which, a third party secretly inserts itself into a two-party conversation to gather or alter information. When MITM malware installs itself onto your computer or network, it gains the ability to spy on & record sensitive data/info.

MITM gains access by → Weak wifi security → Phishing links → Poor user habbits

Eg. An employee logs in to comany system normally. A MITM malware secretly intercepts the login and modifies the information, locking employee out. The attacker can now steal anything.

Concept of Security Principles

Confidentiality and Privacy - only authorized ppl should be able to access sensitive data

Authentication - Checks who you are before allowing access. Verifies the identity of user. Eg. login to Gmail.

Authorization and Access control - Once you are authenticated, the system decides what you are allowed to do. Goal is to make sure users only access the resources they are permitted to. Eg. Student can view marks but not update them, only teachers are authorized to do that.

Integrity, Availability

Non-repudiation - neither the sender, nor the receiver server can deny sending or receiving a message. Goal is to ensure accontability & trust in communication. Eg. Digital signature on an email acts as a proof that the sender cannot deny sending it later.

> Stream Ciphers

What is Cipher? method of converting plain text into unreadable text (ciphertext) to protect the information. It is the core of encryption, used to secure communication. It is a set of algorithms that perform encryption/decryption on a message.

Stream Ciphers is a type of symmetric encryption method where encryption and decryption are done one symbol at a time - usually a character or bit.

P - Plaintext stream
The original message (like Hello)

K - key stream
seq of bits or char used to encrypt

C - Ciphertext stream
The final encrypted output.

$P = P_1, P_2, P_3 \ldots$
$C = C_1, C_2, C_3 \ldots$
$K = (K_1, K_2, K_3 \ldots)$

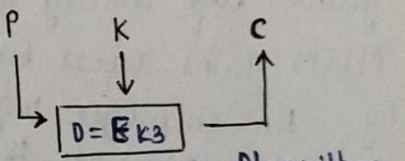
Each bit from P is combined with corresponding bit from K (using XOR operation)
The result is one bit of C (ciphertext).
This happens one symbol at a time (not in blocks)
The same key stream is used to decrypt the ciphertext back to plaintext.

$C_1 = E_{K_1}(P_1)$
$C_2 = E_{K_2}(P_2)$
$C_3 = E_{K_3}(P_3)$
$C_i = P_i \oplus K_i$

Eg. P - 1011
K - 1100
} XOR operation gives C - 0111

To decrypt
C - 0111
K - 1100
} XOR operation gives P - 1011

Useful for real-time applications like voice, video or live chats.



Encryption Algorithm.

> Substitution ciphers

It replaces each symbol (letter or digit) in the plaintext with another symbol.

Types -

1) Mono alphabetic Cipher

Each letter is always replaced with same letter throughout the message. ie, the relationship between letters in plaintext & the ciphertext is one-to-one

Eg. Additive Cipher (Caesar cipher)

If key (K) = 3 : A → D, B → E ..., Z → C So HELLO → KHOOR

Eg. P - all capitals C - all small letters

Additive cipher with K = 15 to encrypt msg 'hello'

| P: | | Encryption E: | C: |
|---|---|---|---|
| h - 7 | | (7+15) mod 26 | W |
| e - 4 | | (4+15) mod 26 | T |
| l - 11 | | (11+15) mod 26 | A |
| l - 11 | | (11+15) mod 26 | A |
| o - 14 | | (14+15) mod 26 | D |

Mono alphabetic ciphers are easy to break using frequency analysis.

2) Polyalphabetic Cipher

Each letter can be replaced by different letters depending on its position in the message. It is one-to-many mapping.

Eg. Autokey Cipher

Key is a sequence, not a single value. first letter uses one key, second uses another, etc.

$P = P_1 P_2 P_3 \ldots \quad C = C_1 C_2 C_3 \ldots \quad K = K_1 K_2 K_3 \ldots$

Encryption: $C_i = (P_i + k_i) \bmod 26$ Decryption: $P_i = (C_i - k_i) \bmod 26$

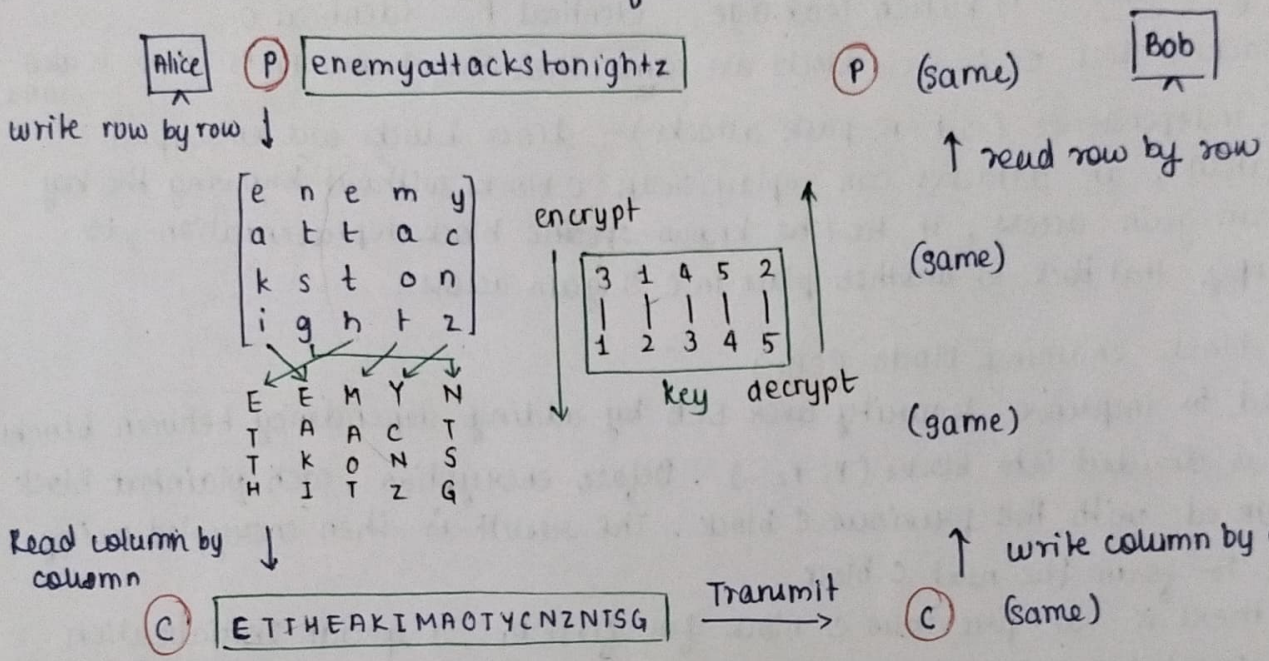
Harder to crack than mono-alphabetic because it hides letter frequency

> Transposition Cipher

This does not substitute one symbol for another, instead it changes the location of the symbols. Basically it reorders symbols

Eg. Message: ENEMY ATTACKS TONIGHT

Steps: Write message in rows. Rearrange the columns using a key (eg 3,1,4 5,2)
Read columns top to bottom for ciphertext.




Rail-Fence technique (zigzag or keyless transposition)

Message is written in zigzag pattern across multiple "rails" (rows) then read row by row to get ciphertext.

Eg. This is a secret msg.

Rail 1 T i i a e r t s
Rail 2 h s s s c e m g

C = Tiiaertshsssceemg

> Block Cipher

A group of plaintext symbols of size m are encrypted together, creating a group of ciphertext of the same size.

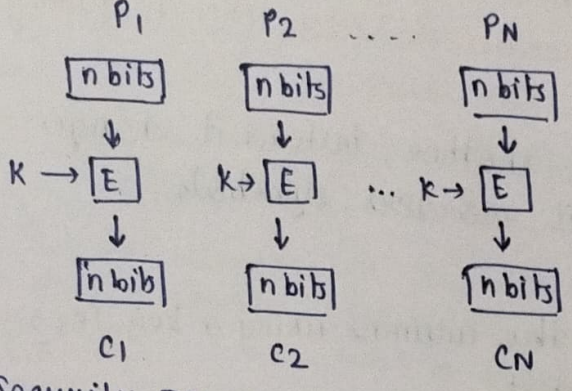
Block Cipher modes

Modes: techniques that describe how blocks of plaintext are encrypted using block cipher

1) ECB - Electronic Code Book

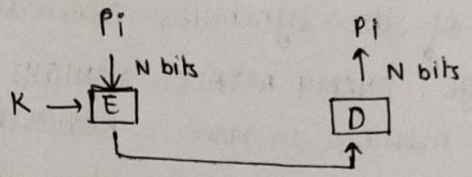
P is divided into N blocks (fixed). The block size is n bits (fixed).
If the P size is not multiple of Block size, then text is padded (makes it complete)

The same encryption key is applied to every block of plaintext. No key changes per block - its fixed and reused.




Encryption $C_i = E_K(P_i)$

Decryption $P_i = D_K(C_i)$




Security Issues i) Patten leakage : identical P = identical C

If attacker sees $C_1, C_5$ & $C_{10}$ blocks are same, then they know $P_1$ $P_5$ & $P_{10}$ is also same.

ii) Block independence (Cut-n-paste attacks) : Since blocks are encrypted separately, an attacker can replay/swap C blocks without knowing the key One can gain access, if he/she knows specific block representation, it can copy that block to another place in C & gain access.

Cipher Block Chaining Mode (CBC)

Designed to improve security over ECB by adding dependency between blocks The P is divided into blocks ($P_1$ $P_2$ ..). Before encryption, each plaintext block is XOR ed with the previous C block. The result is then encrypted using the key to form the next C block.

Since, there is no previous C block for first $P_1$, a special Initialization Vector (IV) is used. IV acts as $C_0$ (dummy block)

eg. $C_1 = E(P_1 \oplus IV)$ $C_2 = E(P_2 \oplus C_1)$ $C_3 = E(P_3 \oplus C_2)$ ....

Encryption $C_0 = IV$

$C_i = E_K(P_i \oplus C_{i-1})$

Decryption $C_0 = IV$

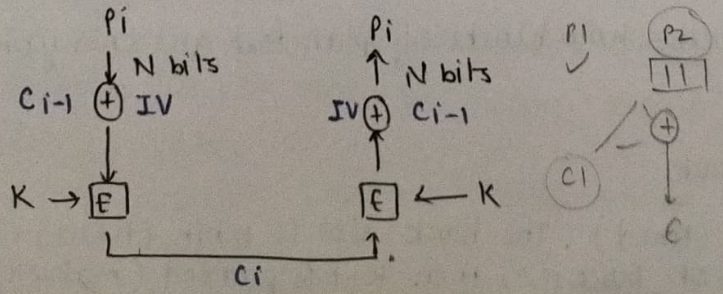
$P_i = D_K(C_i) \oplus C_{i-1}$

Why CBC is more secure?

1) Prevents patten leakage : If $P_1 = P_5$, then $C_1 \neq C_5$
2) Randomness is introduced by using IV
3) Each block depends on previous one.

Drawbacks : Encryption cannot be done in parallel.

One error in C Block affects the current & next block during decryption.

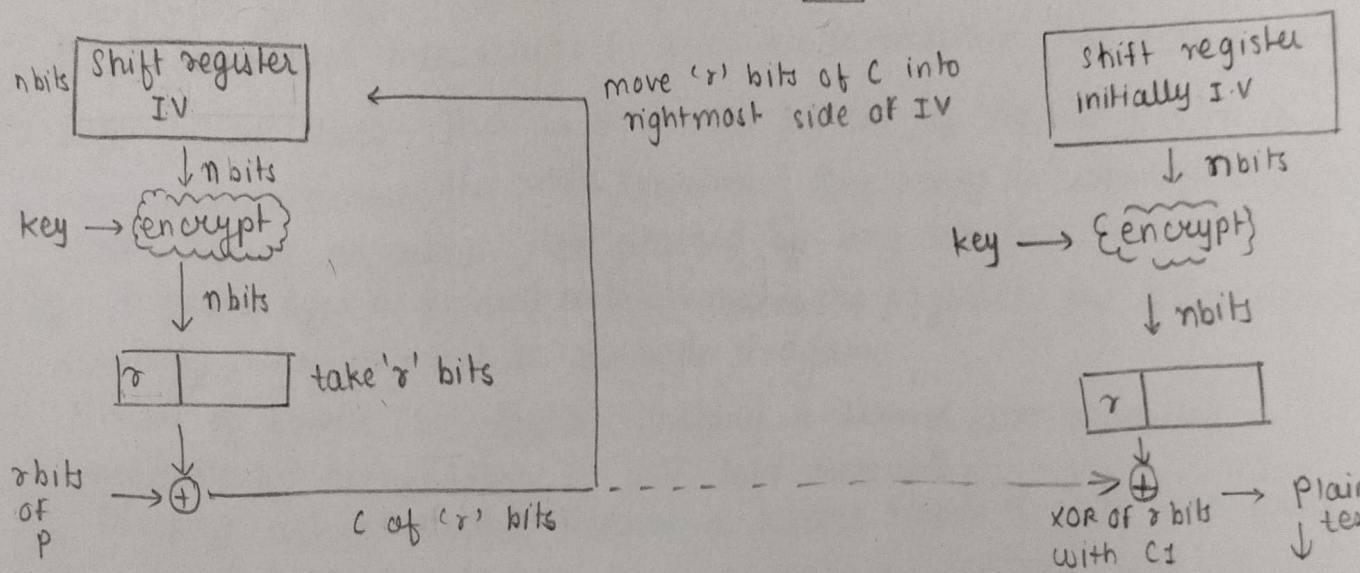


## CFB - Cipher Feedback Mode

It turns block cipher into stream-like cipher.
Lets you encrypt data smaller than the block size.
A shift register (S) of size 'n' bits is used. You encrypt this register - not the plaintext directly. Take only 'r' bits of encrypted result.
XOR those 'r' bits with P block to get C shift this register, and update it with C bits.
Here Both encryption & decryption use encryption function only.

$1 < \text{plaintext size} < n$
$s$

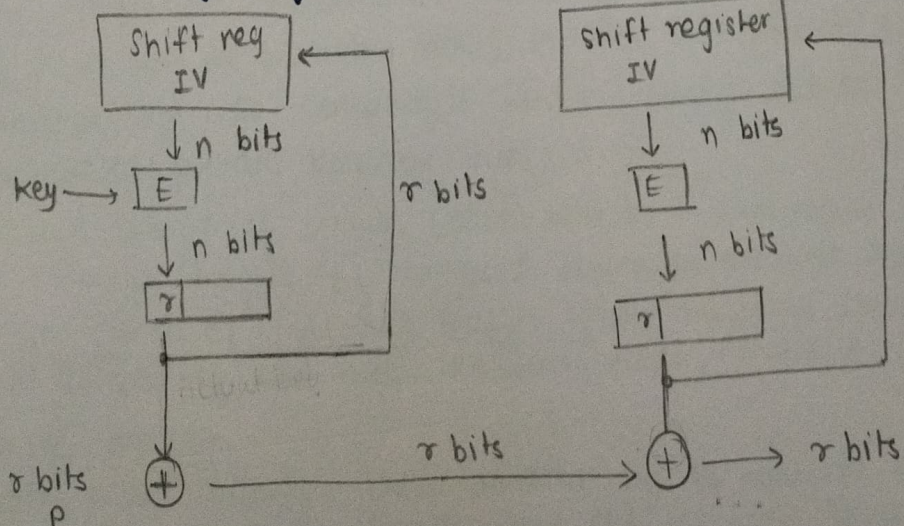
Step 1 (IV) into register that is 'n'
Step 2 → this is encrypted
Step 3 Select s bits from 'n' ie MSB of s-bits
Step 4 XOR we get Ciphertext block




Why to use?
No need to wait for full blocks - can encrypt character by character.

## OFB - Output Feedback mode

Same to CFB, but the output of IV encryption process is fed into the next stage of encryption process




Output 1 = Encrypt(IV)
Ciphertext 1 = Output 1 XOR P1
Output 2 = Encrypt (O/P 1)
C2 = Output 2 XOR P2
:
so on