Software System Security

POOJA PREMNATH

Unit I

Overview of Security and Encryption Techniques

Computer Security Concepts - Security Architecture - Attacks, Services and Mechanisms - Fundamental Security Design Principles - Attack Surfaces and Attack Trees - Algebraic Structures - Modular Arithmetic - Euclid's algorithm - Congruence and matrices - Groups, Rings & Fields - Finite Field block cipher - Data Encryption Standard - Advanced Encryption Standard - Stream Ciphers and RC4 - Cipher Block Modes of Operation.

* **Computer Security** - The protection afforded to an automated information system to attain the applicable objectives of preserving the confidentiality, integrity and availability of information system resources (includes hardware, software, firmware, information/data and telecommunications)

* **CIA Triad**

(1) **Confidentiality** - covers 2 ideas

A. **Data Confidentiality** : assures that private or confidential information is not made available or disclosed to unauthorized individuals.

B. **Privacy** - assures that individuals control or influence what information related to them is collected, stored & used by others

→ A loss of confidentiality is an unauthorized disclosure of information.

② Integrity - The 2 ideas are:

A. Data Integrity - assures that information - both stored and transmitted packets are changed only in a specified and authorized manner.

B. System Integrity - assures that a system performs its intended function in an unimpaired manner, free from deliberate or inadvertant unauthorized manipulation of the system.

③ Availability - assures that a system ~~performs its intended function in an unimpaired manner, free from deliberate or inadvertant~~ works promptly and service is not denied to authorized users.

* Additional Security Concepts

① **Authenticity** - The property of being genuine and being able to be verified and trusted ; confidence in the validity of a transmission, message or message originator.

→ It means to verify users are who they say they are and that each input arriving at the system came from a trusted source.

(2) **Accountability** - The security goal that generates the actions of an entity to be traced uniquely to that entity. This supports non-repudiation, deterrence, fault isolation, intrusion detection and prevention and after action recovery and legal action.

→ Systems keep records of their activities to permit later forensic analysis to trace security breaches / transaction breaches.

* Levels of Impact of Security Breaches on an Organization

A. **Low** - limited adverse effect on organizational operations, assets or individuals

- may cause
  - (i) degradation in mission capability, effectiveness of org. noticeably reduced.
  - (ii) minor damage to organizational assets
  - (iii) result in minor financial loss
  - (iv) result in minor harm to individuals

B. **Moderate** : expected to have a serious adverse effect on organizational operations, organizational assets, individuals

- may cause
  - (i) significant degradation in mission capability, effectiveness of functions is significantly reduced

(ii) result in significant damage to ~~financial~~ organizational assets

(iii) result in significant financial loss

(iv) result in significant harm to individuals that does not involve loss of life or serious life-threatening injuries

c. **High** - severe or catastrophic adverse effect on organizational operations, assets or individuals.

- may cause:

(i) severe degradation or loss of mission capability to an extent that the org. is not able to perform one or more of its primary functions

(ii) result in major damage to org. assets

(iii) result in major financial loss

(iv) result in severe or catastrophic harm involving loss of life / life-threatening injuries

* Challenges of Computer Security

1. Computer Security is not simple as it seems - the CIA triad is too complex.

2. In developing a particular security mechanism, one must always consider potential attacks on those security features.

3. Procedures used to provide particular services are often counterintuitive.

4. Physical & logical placement needs to be determined (Network and TCP/IP Layer)

5. Security mechanisms typically involve more than a particular algorithm or protocol and also require that participants be in possession of some secret information, which raises questions about the creation distribution and protection of that secret information.

6. Attackers only need to find a single weakness, while the designer must find and eliminate all weaknesses to achieve perfect security.

7. Security is still too often an afterthought to be incorporated into a system after the design is complete, rather than it being an integral part of the design process

8. Security requires regular and constant monitoring

9. There is a natural tendency on the part of users and system managers to perceive little benefit from security investments until a security failure occurs.

10. Many users and even security administrators view strong security as an impediment to efficient and user-friendly operation of an information system or use of information

* Computer Security Terminology

(i) Adversary - individual / group / organization that has the intent to conduct detrimental activities

(ii) Attack - any kind of malicious activity that attempts to collect, disrupt, deny, degrade a destroy information system

(iii) Countermeasures - A device/technique that has the objective of impairing the operational effectiveness of undesirable/adversarial activity.

(iv) Risk - A measure of the extent to which an entity is threatened by a potential circumstance or event

(v) Security Policy - A set of criteria for the provision of security services - defines activities to maintain a condition of security for systems and data.

(vi) System Resource (Asset) - A major application, general support system, high impact program, mission critical system, or a logically related group of systems

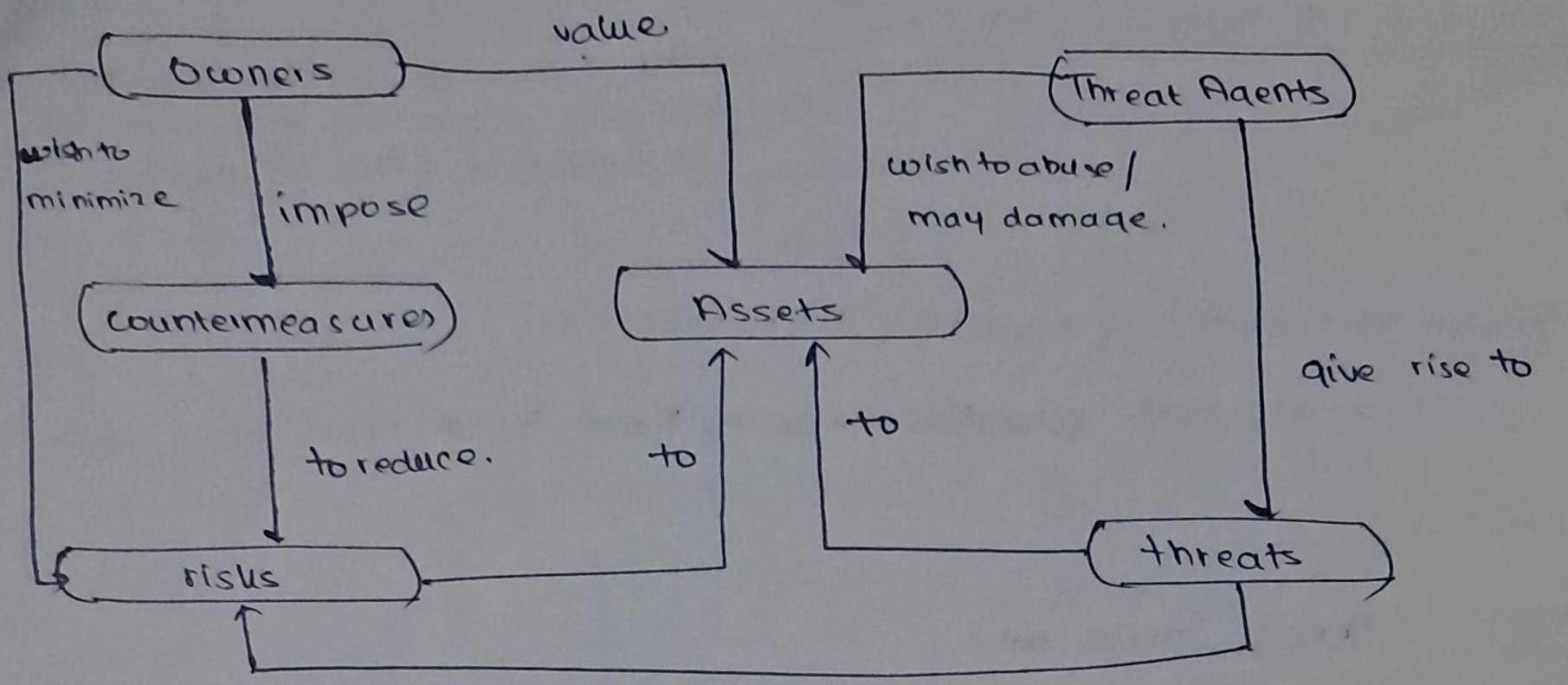
(vii) Threat - any circumstance or event with the potential to adversely impact organizational operations

(viii) Vulnerability - weakness in an information system / implementation that could be exploited.

* Assets of a Computer System

(i) Hardware
(ii) Software
(iii) Data
(iv) Communication facilities and networks

* Security Concepts and Relationships



* Computer and Network Assets with Examples of Threats

	Availability	Confidentiality	Integrity
Hardware	equipment is stolen or disabled, denying service	unencrypted CD ROM / DVD is stolen	
Software	programs are deleted, denying access to users	an unauthorized copy of software is made.	A worm or a program is modified, either causing it to fail during execution or to cause it to do some unintended task
Data	Files are deleted, denying access to users	Unauthorized read of data is performed. Analysis of statistical data reveals underlying data.	Existing files are modified or new files are fabricated.
Communication lines and Networks	Messages are destroyed or deleted. Communication lines or network are rendered unavailable.	Messages are read. The traffic pattern of messages is observed.	Messages are modified, delayed, recorded or duplicated. False msgs are fabricated.

* Vulnerabilities, Threats and Attacks

Categories of Vulnerabilities :
- Corrupted (loss of integrity)
- Leaky (loss of confidentiality)
- Unavailable (loss of availability)

Threats → capable of exploiting vulnerabilities
- represent potential security harm to an asset

Attacks (threats carried out)

Types:

(i) Passive - attempt to learn or make use of information from the system that does not affect system resources

(ii) Active - attempt to alter system resources or affect their operation

(iii) Insider - initiated by an entity inside the security parameter.

(iv) Outsider - initiated from outside the perimeter

Threats, Consequences and the Threat Actions

Threat Consequence	Threat Action (Attack)
(1) Unauthorized disclosure	(i) exposure (ii) interception (iii) inference (iv) intrusion
(2) Deception	(i) masquerade (ii) falsification (iii) repudiation
(3) Disruption	(i) Incapacitation (ii) Corruption (iii) Obstruction
(4) Usurpation	(i) Misappropriation (ii) misuse

* The OSI Security Architecture

→ X.800 = a security architecture for OSI

→ The OSI architecture focuses on:

(i) Security Attacks - any action that compromises the security of information owned by an organization.

(ii) Security Mechanism - a process or device that is designed to detect, prevent or recover from a security attack

(iii) Security Service - a processing or communication service that enhances the security of the data processing systems and the information transfers of an organization.

A. Security Attacks

→ includes passive and active attacks

Passive Attacks

→ eavesdropping or monitoring of transmissions

→ opponent aims to obtain info that is being transmitted.

→ Two types of passive attacks:

(i) Release of message contents

(ii) Traffic analysis - observe pattern of messages, and determine the location & identity of hosts, and observe frequency and length of messages.

Active Attacks

→ involve some modification of the data stream or the creation of a false stream

→ can be divided into 4 categories

(i) Masquerade - one entity pretends to be a different entity, helps capture authentication sequences and obtain extra priveleges

(ii) Replay - passive capture of a data unit & subsequent retransmission

→ difficult to detect passive attacks, since there is no alteration of data.

→ Neither sender nor receiver is aware that a third party has read the messages or observed the traffic pattern.

→ Prevent these attacks using encryption

(iii) Modification of messages - means that some portion of a legitimate msg is altered, or msgs are delayed or reordered

(iv) Denial of Services - prevent or inhibits the normal use or management of communication facilities. The attack may have a specific target or disrupt the entire network

→ Difficult to prevent active attacks because of the wide variety of potential physical, software & network vulnerabilities

B. Security Services

→ X.800 defines a security service as a service that is provided by a protocol layer of communicating open systems and that ensures adequate security of the systems or data transfers.

→ RFC 4949 defines security service as a processing or communication service that is provided by a system to give a specific kind of protection to system resources

X.800 divides security services into 5 categories & 14 specific services

(1) **Authentication** → The assurance that the communicating entity is the one that it claims to be.

A. Peer Entity Authentication - used in association with a logical connection to provide confidence in the identity of the entities connected.

B. Data-Origin Authentication - In a connectionless transfer, provides assurance that the source of retrieved data is as claimed.

(2) Access Control - prevention of the unauthorized use of a resource - i.e this service contds who can have access to a resource, under what conditions access can occur, and what those accessing the resources are allowed to do.

(3) Data Confidentiality - The protection of data from unauthorized disclosure.

A. Connection Confidentiality - protection of all user data on a connection

B. Connectionless Confidentiality - the protection of all user data in a single data block.

C. Selective-Field Confidentiality - The confidentiality of selected fields within the user data on a connection or in a single data block.

D. Traffic Flow Confidentiality - The protection of the information that might be derived from observation of traffic flows.

(4) Data Integrity - The assurance that data received are exactly as sent by an authorized entity - i.e contains no modification, insertion,

deletion or replay.

A. Connection Integrity with Recovery - provides for the integrity of all user data on a connection and detects any modification, insertion, deletion or replay of any data within an entire data sequence, with recovery attempted.

B. Connection Integrity without Recovery - same as A. but provides only detection without recovery

C. Selective Field Connection Integrity - provides for the integrity of selected fields within the user data of a data block transferred over a connection and takes the form of determination of whether the selected fields have been modified, inserted, deleted or replayed.

D. Connectionless Integrity - provides for the integrity of a single connectionless data block and may take the form of detection of data modification. Additionally, a limited form of replay detection may be provided.

E. Selective Field Connectionless Integrity - provides for the integrity of selected fields within a single connectionless block, takes the form of determination whether the selected fields have been modified.

(5) Non-Repudiation → provides protection against denial by one of the entities involved in a communication of having participated in all or part of the communication

A. Non Repudiation - Origin - proof that the message was sent by the specified party

B. Non-Repudiation-Destination - proof that the message was received by the specific party.

C. X.800 Security Mechanisms

Specific Security Mechanisms	Pervasive Security Mechanisms
→ May be incorporated into the appropriate protocol layer in order to provide some of the OSI security services. ① Encipherment - use of mathematical algorithms to transform data into a form that is not readily intelligible. The transformation and subsequent recovery depends on an algorithm & 0 or more encryption keys. ② Digital Signature - Data appended to a data unit that allows a recipient to prove the source and integrity of the data unit and protect against forgery ③ Access Control - a variety of mechanisms that enforce access rights to resources ④ Data Integrity	→ Mechanisms that are not specific to any particular OSI security service or protocol layer. ① Trusted Functionality - that which is perceived to be correct w.r.t to some criteria ② Security Label - the marking bound to a resource that names or designates the security attributes of that resource ③ Event Detection - detection of security-related events. ④ Security Audit Trail - data collected for a security audit - an independent review & examination of system records and activities ⑤ Security Recovery - deals w/ requests from mechanisms

Specific Security Mechanisms (contd.)

(5) Authentication Exchange - a mechanism intended to ensure the identity of an entity by means of info. exchange

(6) Traffic Padding - the insertion of bits into gaps in a data stream to frustrate traffic analysis attempts

(7) Routing Control - enables selection of particular physically secure routes for certain data and allows routing changes

(8) Notarization - the use of a trusted third party to assure certain properties of a data exchange.

* Relationships between Security Services and Mechanisms

	Encipherment	Digital Signature	Access Control	Data Integrity	Authentication Exchange	Traffic Padding	Routing Control	Notarization
Peer Entity Authentication	✓	✓			✓			
Data Origin Authentication	✓	✓						
Access Control			✓					
Confidentiality	✓						✓	
Traffic Flow confidentiality	✓					✓	✓	
Data Integrity	✓	✓		✓				
Non-repudiation		✓		✓				✓
Availability				✓	✓			

* Fundamental Security Design Principles

(1) Economy of mechanism - design of security measures embodied in both hardware and software should be as simple and small as possible.

→ Relatively simple designs are easier to test and verify thoroughly.

→ With a complex design, there are many opportunities for an adversary to discover subtle weaknesses.

(2) Fail-Safe Defaults - The default situation is lack of access, and the protection scheme identifies conditions under which access is permitted.

(3) Complete Mediation - every access must be checked against the access control mechanism. Systems should not rely on access decisions retrieved from a cache.

(4) Open Design - the design of a security mechanism should be open rather than secret. The algorithms can then be reviewed by many experts, and users can have high confidence in them.

(5) Separation of Privilege - multiple privilege attributes are needed to achieve access to a restricted resource. eg. MFA

(6) Least privileges - each process and every user of the system should operate using the least set of privileges necessary

eg. role-based access control. This system security policy can identify and define the various roles of users or processes.

(7) Least Common Mechanism - the design should minimize the functions shared by different users, providing mutual security.

(8) Psychological Acceptability - security mechanisms should not unduly interfere with the work of users, while at the same time meeting the needs of those who authorize access.

If security mechanisms hinder the accessibility or usability of resources users may opt to turn off those mechanisms.

(9) Isolation

→ Public access systems should be isolated from critical resources to prevent disclosure or tampering.

→ Physical isolation may include ensuring that no physical connections exist between an organization's public info. resources and an organization's critical systems.

→ The processes and files of individual users should be isolated from one another except when it is explicitly desired.

→ Security mechanisms must also be isolated to prevent access to them

(10) Encapsulation - a specific form of isolation based on object-oriented functionality. Protection is provided by encapsulating a collection of procedures and data objects in a domain of its own.

(11) Modularity - Develop security functions as separate, protected modules and use a modular architecture for mechanism design and implementation.

(12) Layering - the usage of multiple, overlapping protection approaches.

→ provides multiple barriers between an adversary and protected information or services., technique is called defense in depth

(13) Least Astonishment - a program or user interface should always respond in the way that it is least likely to astonish the user. For eg, the mechanism for authorization should be transparent enough that the user has a good / intuitive understanding of how the security goals map to the provided security mechanism.

* Attack Surfaces and Attack Trees

A. Attack Surfaces

→ consist of the reachable and exploitable vulnerabilities of a system.

Examples

(i) open ports on outward facing web and other services and code listening on those ports

(ii) surfaces available on the inside of a firewall

(iii) code that processes incoming data, email, XML, office documents and industry specific custom data exchange formats

(iv) Interfaces, SQL & web forms

(v) An employee with access to sensitive information vulnerable to a social engineering attack.

Categories of Attack Surfaces

(i) Network Attack Surfaces

→ Vulnerabilities over an enterprise network, wide area network a the Internet

→ includes network protocol vulnerabilities, such as those used for a DoS attack, disruption of communication links and various forms of intruder attacks.

(ii) Software Attack Surfaces

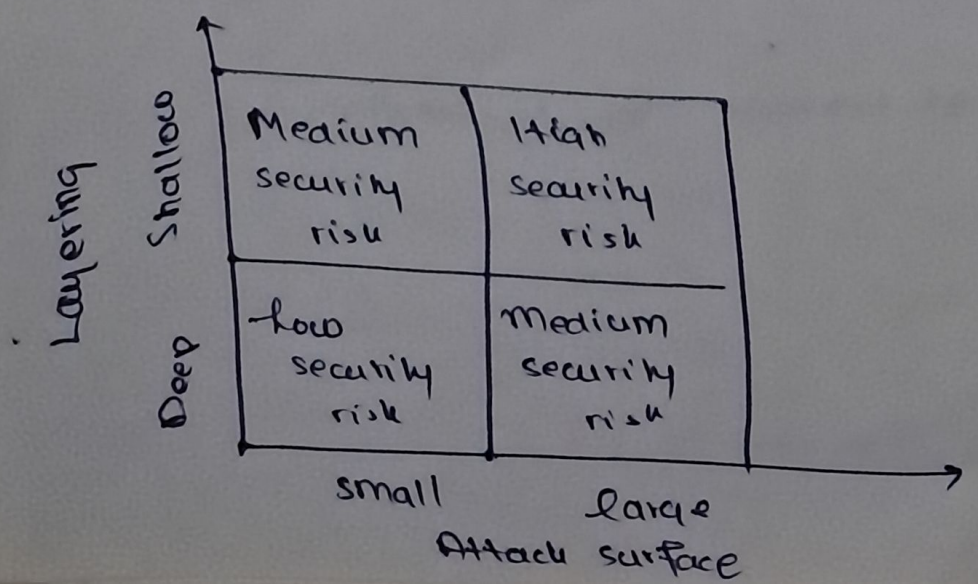
→ vulnerabilities in application, utility or operating system code

→ particular focus on web server software

(iii) Human Attack Surface

→ vulnerabilities created by personnel or outsiders, such as social engineering, human error and trusted insiders

→ The use of layering or defense in depth & attack surface reduction complement each other in mitigating security risk.



B. Attack Trees

→ An attack tree is a branching, hierarchical data structure that represents a set of potential techniques for exploiting security vulnerabilities.

→ The security incident that is the goal of the attack is represented as the root node of the tree, and the ways that an attacker could reach the goal are iteratively and incrementally represented as subnodes and branches of the tree.

→ Each subnode defines a subgoal, each each subgoal may have it own set of further subgoals etc.

→ The final nodes, i.e the leaf nodes, represent different ways to initiate an attack.

→ Each node other than a leaf is either an AND-node or OR-node.

→ Attack trees effectively exploit the information available on attack patterns.

→ Security analysts can use the attack tree to document security attacks in a structured form that reveals key vulnerabilities.

* Mathematics of Symmetric Key Cryptography?

Modular Arithmetic

→ If a is an integer and n is a positive integer, a mod n is the remainder when a is divided by n, ~~the~~ ~~qat~~

$$a = qn + r$$

For negative numbers:

(-ve no) modulo k = k minus positive no. modulo k

i.e $\boxed{(-n)\%k = k - (n\%k)}$

For eg. 11 mod 7 = 4

-11 mod 7 = 7 - (11 mod 7)

= 3

Congruent modulo n - Two integers a and b are said to be congruent modulo n if (a mod n) = (b mod n)

This is written as $a = b \pmod n^2$

| Note: $a = b \pmod n$ if $n(a-b)$ |

For eg

(i) 23 = 8 mod 5

here a = 23

b = 8

a - b = 15

15 divides 5

Illy (ii) -11 = 5 (mod 8)

a = -11

b = 5

a - b = -16

-16 divides 8

Additional Modular Arithmetic Properties

$$[(a \bmod n) + (b \bmod n)] \bmod n = (a+b) \bmod n$$

$$[(a \bmod n) - (b \bmod n)] \bmod n = (a-b) \bmod n$$

$$[(a \bmod n) * (b \bmod n)] \bmod n = (a*b) \bmod n$$

For eg. $11 \bmod 8 = 3$

$15 \bmod 8 = 7$

$$[(11 \bmod 8) + (15 \bmod 8)] = (26) \bmod 8$$

$$= 2$$

Relatively Prime - Two integers are relatively prime if their only common positive integer factor is 1.

Euclid's Algorithm

$$\boxed{gcd(a,b) = gcd(b, a \bmod b)}$$

Eg! $gcd(55, 22)$

Q	A	B	R
2	55	22	11
2	22	11	0
	11	0	

$\boxed{gcd = 11}$

eg? gcd (252, 105)

Q	A	B	R
2	252	105	42
2	105	42	21
2	42	21	0
	21	0	

→ gcd

$105\overline{)252}\ 2$ — 210 — 42

$42\overline{)105}\ 2$ — 84 — 21

A mod B.

$A - A \div B \times B$

$- \div x$

Extended Euclid's Algorithm

→ The extended Euclid's algorithm also finds integer coefficients x and y such that $\boxed{ax + by = \gcd(a, b)}$

example1 : Find the GCD of (161, 28) using the extended euclid's algorithm and find s & t

Note: (A > B)

Q	A	B	R	s_1	s_2	s	t_1	t_2	t
5	161	28	21	1	0	1	0	1	-5
1	28	21	7	0	1	-1	1	-5	6
3	21	7	0	1	-1	4	-5	6	-23
	(7)	0		(-1)	4		(6)	-23	

$s_1 = 1 \quad s_2 = 0$

$t_1 = 0 \quad t_2 = 1$

$s = s_1 - s_2 \times q$

$t = t_1 - t_2 \times q$

$\boxed{\gcd = 7}$

equation

$\boxed{7 = 161(-1) + 28(6)}$

Example2 gcd of 65, 40 using the EEA and find s2t.

Q.	A	B	R	s_1	s_2	s	t_1	t_2	t
1	65	40	25	1	0	1	0	1	-1
1	40	25	15	0	1	-1	1	-1	2
1	25	15	10	+1	-1	2	-1	2	-3
1	15	10	5	-1	2	-3	2	-3	5
2	10	5	0	2	-3	8	-3	5	-13
	(5)	0		(-3)	8		(5)	-13	

$s = s_1 - s_2 \times q$

$t = t_1 - t_2 \times q$

gcd = 5

eqn

$5 = (-3)(65) + (40)(5)$.

(i) $s = 1 - (0)(1)$
$= 1$

$t = 0 - (1)(1)$
$= -1$

(ii) $s = 0 - (1)(1)$
$= -1$

$t = 1 - (-1)(1)$
$= 2$

(iii) $s = 1 - (-1)(1)$
$= 2$

$t = -1 - (2)(1)$
$= -3$

(iv) $s = -1 - (2)(1)$
$= -3$

$t = 2 - (-3)(1)$
$= 5$

(v) $s = 2 - (-3)(2)$
$= 8$

$t = -3 - (5)(2)$
$= -3 - 10 = -13$

Example 3 : Find the inverse of 15 mod 26

Inv = t1

Q	A	B	R	t_1	t_2	t
1	26	15	11	0	1	-1
1	15	11	4	1	-1	2
2	11	4	3	-1	2	-5
1	4	3	1	2	-5	7
3	3	1	0	-5	7	-26
	(1)	0		(7)	-26	

(i) $t = 0 - (1)(1) = -1$

(ii) $t = 1 - (-1)(1)$
$= 2$

(iii) $t = -1 - (2)(2)$
$= -1 - 4 = -5$

(iv) $t = t = 2 - (-5)(1)$
$= 2 + 5 = 7$

(v) $t = -5 - (7)(3)$
$= -5 - 21 = -26$

Inverse = 7

Example 5 : Find the multiplicative inverse of 10 mod 11

Q	A	B	R	s_1	s_2	s	t_1	t_2	t
1	11	10	1	1	0	1	0	1	-1
10	10	1	0	0	1	-10	1	-1	11
	1	0		1	-10		-1	11	

(i) $s = 1 - (0)(1)$
$= 1$
$t = 0 - (1)(1)$
$= -1$

(ii) $s = 0 - (1)(10)$
$= -10$
$t = 1 - (-1)(10)$
$= 11$

gcd = 1

inverse = $t_1 = -1$

to remove the -ve : $t_1 + A$ ← original

$= -1 + 11$

$= 10$

MI = 10

Check like this

MI of 10 mod 11 is 10

here $(10 \times 10) \% 11 = 100 \% 11$

$= 1$

*Groups, Rings and Fields

Groups – A set of elements with a binary operation denoted by • that associates to each ordered pair (a, b) of elements in G an element $a \cdot b$ in G, such that the following axioms are obeyed:

(i) Closure : $a, b \in G$, then $a \cdot b \in G$

(ii) Associativity : $(a \cdot b) \cdot c = a \cdot (b \cdot c)$

(iii) Identity : There exists an element e in G such that $a \cdot e = e \cdot a = a \ \forall a \in G$

(iv) Inverse : $\forall a \in G$, $a \cdot a^{-1} = a^{-1} \cdot a = e$

(v) Commutative : $a \cdot b = b \cdot a$, $\forall a, b \in G$ (abelian group)

Cyclic Group : Exponentiation is defined within a group as a repeated application of the group operator, so that $a^3 = a \cdot a \cdot a$

→ $a^0 = e$

→ $a^{-n} = (a^1)^n$, where a^1 = inverse of the element

→ A group is said to be cyclic if every element of G is a power of a^k (k is an integer) of a fixed element.

→ The element a is said to generate the group G or to be a generator of G.

→ A cyclic group is always abelian and may be finite or infinite

Rings - denoted as $\{R, +, *\}$, satisfies the following axioms

1. All the properties of a group
2. Closure under multiplication: If $a, b \in R$, $ab \in R$
3. Associativity under multiplication: $a(bc) = (ab)c \quad \forall a, b, c \in R$
4. Distributive law : $a(b+c) = ab + ac$ and $(a+b)c = ac + bc$

   $\forall a, b, c \in R$
5. Commutativity of multiplication: $ab = ba \quad \forall a, b \in R$
6. Multiplicative Identity $a1 = 1a = a \quad \forall a \in R$
7. No zero divisors : If $a, b \in R$ and $ab = 0$, then either $a = 0$ or $b = 0$

→ called integral domain / commutative ring

Fields - denoted as $\{F, +, *\}$

→ all properties of groups & rings and:

(i) Multiplicative Inverse - $aa^{-1} = (a^{-1})a = 1$

Field
- Integral Domain
  - Commutative Ring
    - Ring
      - Abelian Group
        - Group

(A1) Closure under addition: $a, b \in S$, $a+b \in S$

(A2) Associativity under addn: $(a+b)+c = a+(b+c)$ $\forall a, b, c \in S$

(A3) Additive Identity: $a+0 = 0+a = a, \forall a \in S$

(A4) Additive Inverse: $a+(-a) = (-a)+a = 0$

(A5) Commutativity of addition: $a+b = b+a$ $\forall a, b \in S$

(M1) Closure under multiplication: $a, b \in S$, $ab \in S$

(M2) Associativity under multiplication: $a(bc) = (ab)c$ $\forall a, b, c \in S$

(M3) Distributive laws: $(a+b)c = ac+bc$ $\forall a, b, c \in S$

(M4) Commutativity of multiplication: $ab = ba$ $\forall a, b \in S$

(M5) Multiplicative identity: $a1 = 1a = a$ $\forall a \in S$

(M6) No zero divisors: If $a, b \in S$ $ab = 0$ then either $a = 0$ or $b = 0$

(M7) Multiplicative Inverse: If $a \in S$ and $a \neq 0$ $aa^{-1} = a^{-1}a = 1$

* Congruence Modulo m

Two integers a and b are congruent modulo if they have the same remainder when divided by m

denoted by: $a \equiv b \pmod m$

$\Rightarrow a \bmod m = b \bmod m$

$[4]_y$
$\Rightarrow x \equiv 4 \pmod y$

Eg1 Find the equivalence class of 2 wrt to congruence modulo 5

$\Rightarrow x \bmod 5 = 2 \bmod 5$

$2 \bmod 5 = 2$

$[2] = \{ \ldots 2 \ldots \}$

Add and subtract 5 on both sides

$[2] = \{ \ldots -8, -3, 2, 7, 12 \ldots \}$

* Modulus of Negative Numbers

$$-a \bmod b = b - (a \bmod b)$$

eg1 $-6 \bmod 5 = 5 - (6 \bmod 5)$

$= 5 - 1$

$= 4$

eg2 $-27 \bmod 10 = 10 - (27 \bmod 10)$

$= 10 - 7$

$= 3$

eg Check if the following congruences are valid.

(i) $15 \equiv 3 \pmod{12}$

$15 \% 12 = 3 \Rightarrow$ valid

(ii) $33 \equiv 3 \pmod{10}$

$33 \% 10 = 3 \Rightarrow$ valid

(iii) $10 \equiv -2 \bmod 12$

$(-2 + 12 = 10)$

~~$a \equiv b \bmod m$~~

$10 \% 12 = 10$

$12 \overline{)\,10}$ (0)

10

(iv) $-8 \equiv 7 \bmod 5$

$-8 \bmod 5 = 5 - (8 \bmod 5)$

$= 5 - 3$

$= 2$

$2 + 5 = 7 \Rightarrow$ valid

alternatively,
do $a - b$
in
$a = b \bmod c$
check if $a - b$ divides c

(v) $-3 \equiv -8 \pmod 5$

~~$-3 \bmod 5 = 5 - (3 \bmod 5)$~~

~~$= 5 - 3$~~

~~$= 2$~~

$-8 - 7$

$= -15$ divides 5

$\Rightarrow$ valid

$-3 \equiv -8 \pmod 5$

$a = -3 \quad b = -8$

$a - b = 5$

5 divides 5 $\Rightarrow$ valid

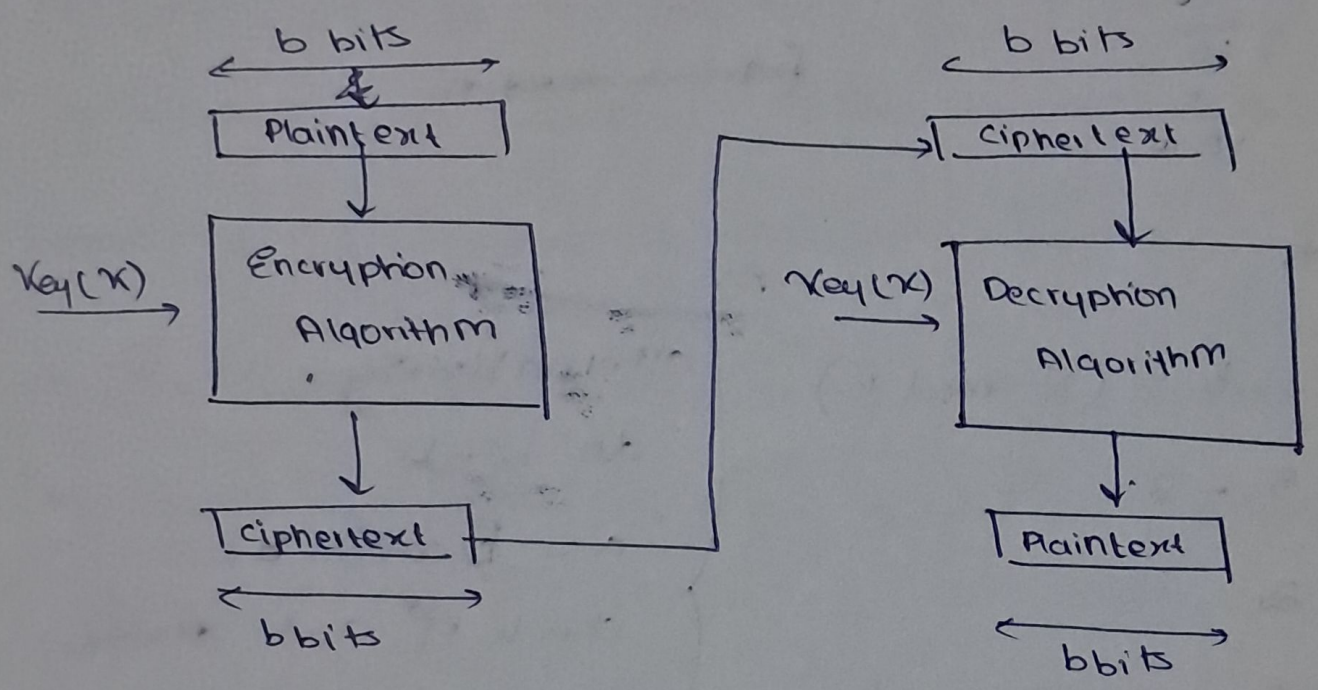
* Finite Field Block Ciphers

* Block Cipher Principles

→ A block cipher is one in which a block of plaintext is treated as a whole and is used to produce a ciphertext block of equal length.

→ Typically, a block size of 64 or 128 bits is used.

→ The two users share a symmetric encryption key.



→ most symmetric block encryption algorithms are based on a structure called the Fiestel block cipher.

* Fiestel Cipher

→ utilizes the concept of a product cipher. - where 2 or more simple ciphers are used in sequence, in such a way that the final product is cryptographically stronger than any of the component ciphers.

→ develops a cipher with a key length of k bits and block length of n bits, allowing a total of 2^k possible transformation

→ The Fiestel cipher alternates substitutions and permutations :

(i) Substitution - Each plaintext element or group of elements is uniquely defined by a corresponding ciphertext element or group of elements.

(ii) Permutation - A sequence of plaintext elements is replaced by a permutation of that sequence. i.e no elements are added or deleted/ replaced in the sequence, rather the order in which the elements appear in the sequence is changed.

→ The Fiestel cipher is an application of a proposal by Claude Shannon to develop a product cipher that alternates confusion and diffusion functions.

* Claude Shannon's ideas of Confusion and Diffusion

Diffusion - dissipates the statistical structure of plaintext over the bulk of cipher text

Confusion - makes relationship between ciphertext and key as complex as possible.

Motivation - To thwart cryptanalysis based on statistical analysis. If the attacker has some knowledge of the statistical characteristics of the plaintext. The frequency distribution of letters / usage of words & phrases may allow one to deduce the encryption key.

* Fiestel Cipher Structure

Encryption

Inputs → (i) plaintext block of length 2w
(ii) Key K

→ Divide the plaintext block into 2 halves LE_0, RE_0

→ Pass each half through n rounds of processing, combine to produce the ciphertext.

Rounds

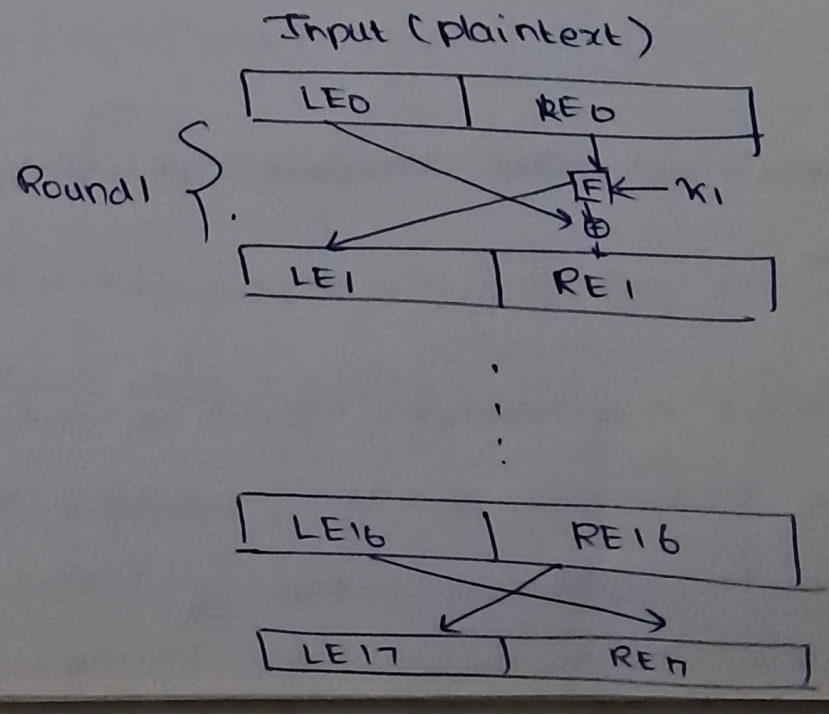
→ Each round has input LE_{i-1} and RE_{i-1} derived from the previous round as well as subkey K_i derived from K.

→ Each round has

(i) Substitution - on LHS of data
→ apply a round function to RHS & take XOR with LHS

(ii) Permutation - interchange 2 halves of data

SPN
proposed by Shannon

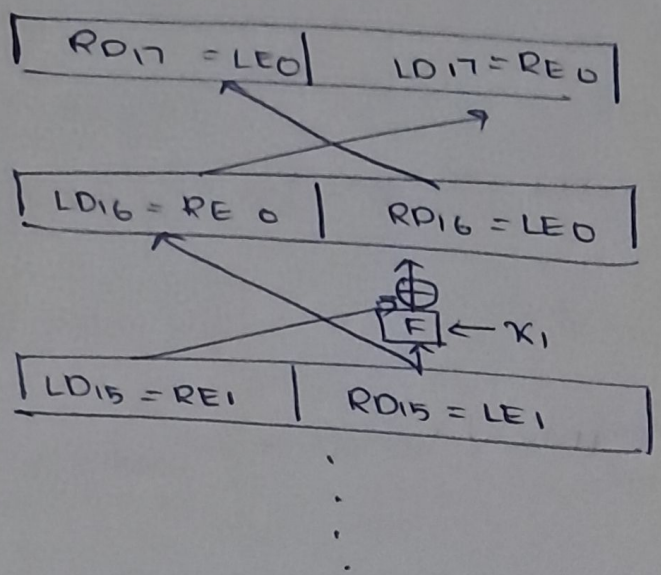


Decryption

Input

→ use ciphertext as input

→ use subkeys K_i in reverse order



* Fiestel Network Design Features

→ The exact realization of a Fiestel network depends on the choice of the following parameters:

① Block size — larger block size ⇒ greater security, but reduced speed

— 64 / 128 bit block sizes

② Key size — larger key ⇒ greater security, decreases speed

— greater security is achieved by greater resistance to brute-force attacks and greater confusion

— 128 bits keys commonly used.

③ Number of rounds - multiple rounds increase security

- a typical ~~cipher~~ size is ~~of size~~ 16 rounds

④ Subkey generation algorithm - greater complexity ⇒ greater difficulty of cryptanalysis

⑤ Round function F - greater complexity ⇒ greater resistance to cryptanalysis.

Other factors: (i) Fast software encryption/decryption

(ii) ease of analysis

* Data Encryption Standard (DES) Algorithm

Features

→ most widely used block cipher in the world.

→ encrypts 64-bit data using 56-bit key

→ used in financial applications, and is still the standard for legacy application use.

DES Encryption

Inputs : plaintext and key

↓ ↓

64 bits 56 bits.

Processing plaintext (i) pass plaintext through initial permutation to produce the permuted input

(ii) 16 rounds of the same function - has both permutation and substitution

→ The output of the last round (16th round) consists of 64 bits that are a function of the input plaintext and key.

→ The left and right halves are ~~passed~~ swapped to produced the pre-output.

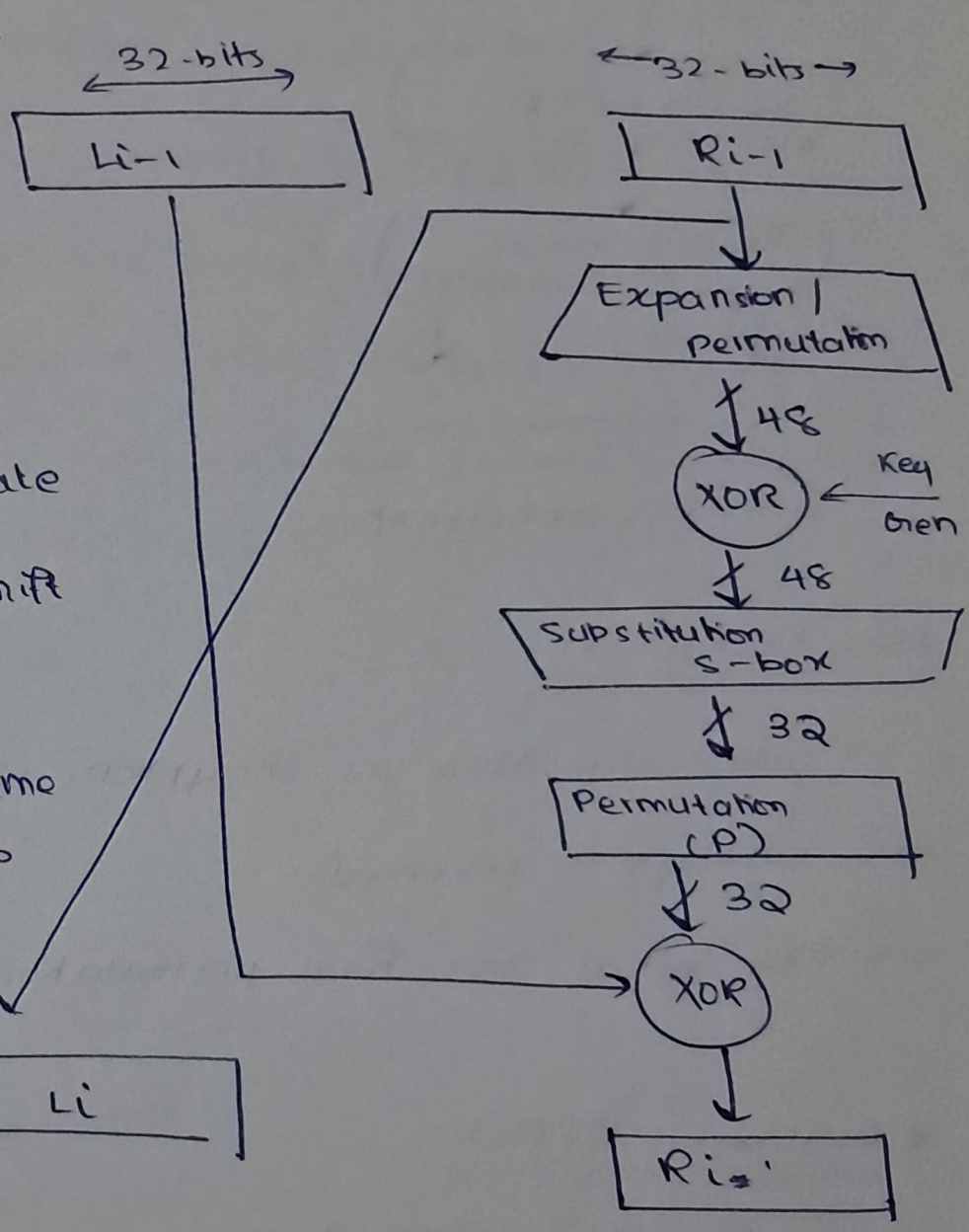
→ The pre-output is passed through a permutation [IP⁻¹] that is the inverse of the initial permutation function to produce the 64-bit ciphertext.

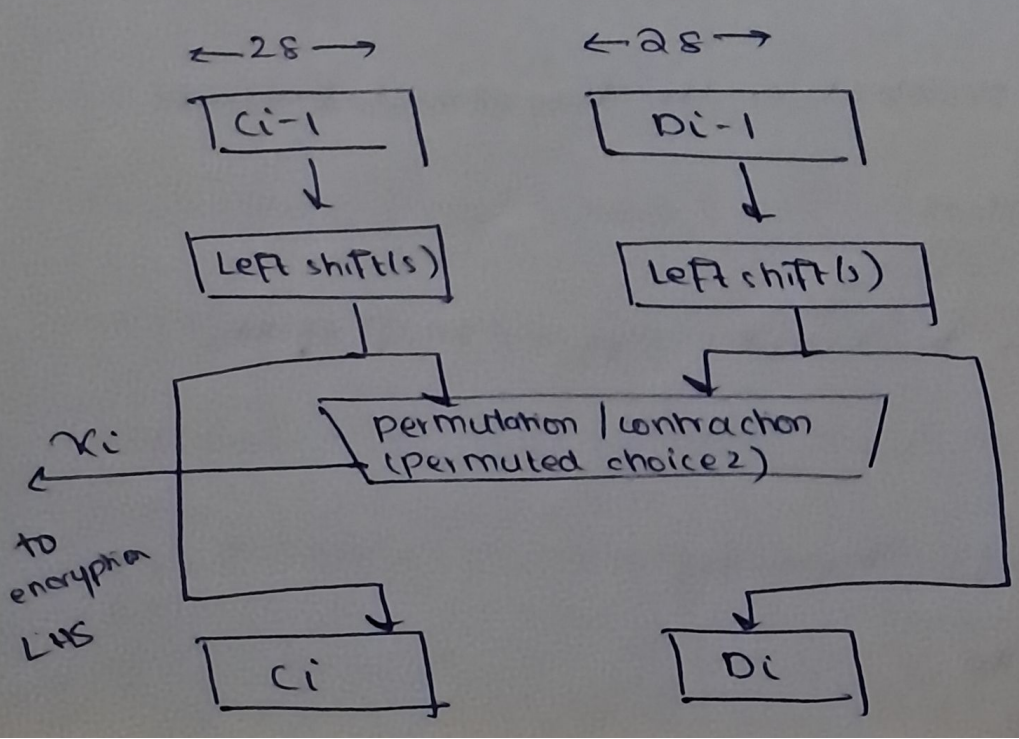
Processing the key

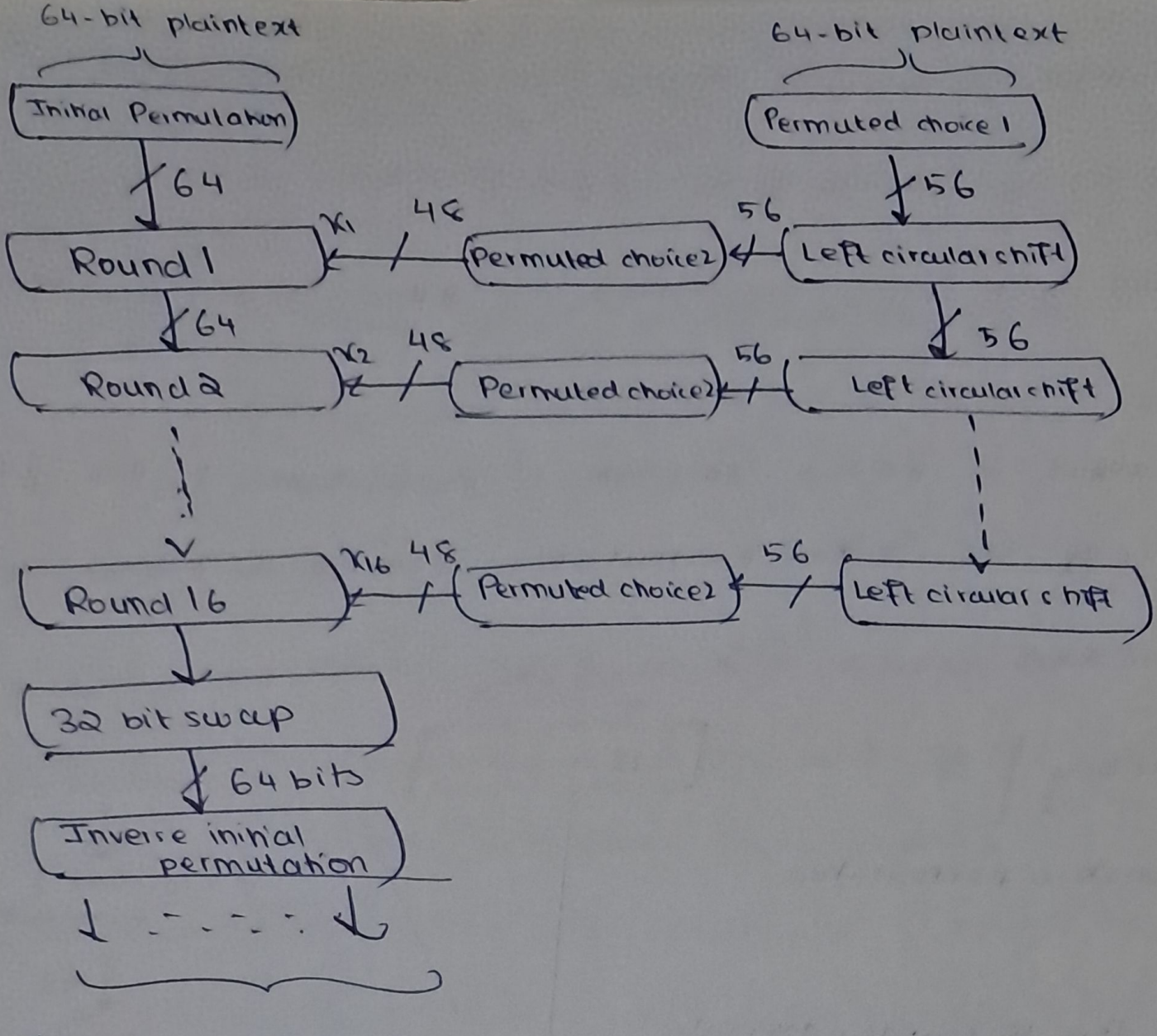
→ pass key through a permutation function

→ for each of the 16 rounds, generate a subkey, from a left-circular shift and a permutation.

→ The permutation function is the same for each round, but a different subkey is produced each time because of the repeated shifting.







DES Decryption

→ same algorithm as encryption, except that the application of the subkeys is reversed.

→ The initial and final permutations are reversed.

* Avalanche Effect

→ a small change in either the plaintext or the key should produce a significant change in the ciphertext.

→ a change of one input results in changing approx half of the output bits

→ makes attempts to guess keys impossible

→ DES exhibits strong avalanche

* Shength of DES

A. Key Size

→ 56-bit keys have $2^{56} = 7.2 \times 16^{16}$ key values

→ brute force looks hard, but can be done in around 22 hrs

B. Analytic Attacks

→ now there exists several analytic attacks on DES

→ these utilize some deep cipher structure of the cipher

(i) gather info about the encryptions

(ii) can eventually recover some / all of the sub-key bits

(iii) if necessary, can exhaustively search for the rest

→ usually are statistical attacks

(i) differential cryptanalysis

(ii) linear cryptanalysis

(iii) related key attacks

C. Timing Attacks

→ attacks actual implementation of cipher

→ use knowledge of consequences of implementation to derive info about some / all subkeys

→ specifically uses the fact that calculations can take varying times depending on the value of inputs to it

→ particularly problematic on smartcards

* Block Cipher Design Principles

A. Number of rounds - greater the no. of rounds, more difficult it is to perform cryptanalysis

B. Design of Function F - F should be non-linear ⇒ increases confusion

Other criteria - (i) good avalanche properties -

strict avalanche criterion

bit independence criterion

C. Key Schedule Algorithm - generate one subkey for each round -

select subkeys to maximize the difficulty of deducing individual subkeys and the difficulty of working back to the main key.

* DES Weaknesses

(i) weaknesses in S-boxes → 2 specifically chosen inputs to an S-box array can create the same output

→ obtain the same output in a single round by changing bits in only 3 neighboring S-boxes

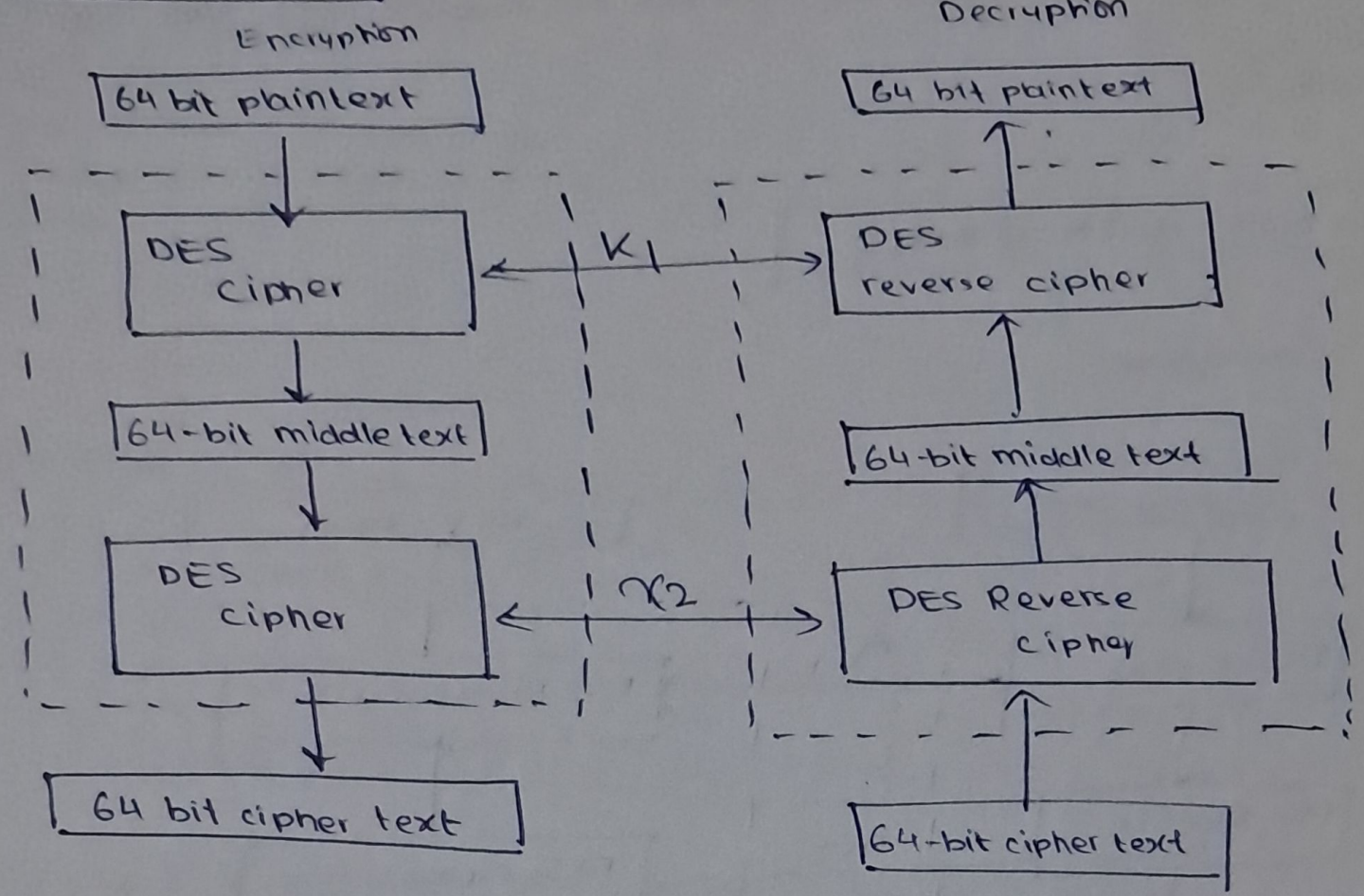
* Desirable Block Cipher Properties

1. Avalanche effect
2. Completeness - each bit of plaintext needs to depend on many bits of the plaintext

(P & S-boxes produce diffusion & confusion to establish completeness)

* Variants of DES

(1) Double DES



→ does not have a key strength of 2^{112} because of man-in-the-middle-attack

Man-in-the-middle Attack

→ Consider the plaintext P and encryption keys K_1 and K_2, a ciphertext C is produced as $C = E_{K_2}(E_{K_1}(m))$

→ Decryption would be $P = D_{K_1}(D_{K_2}(C))$

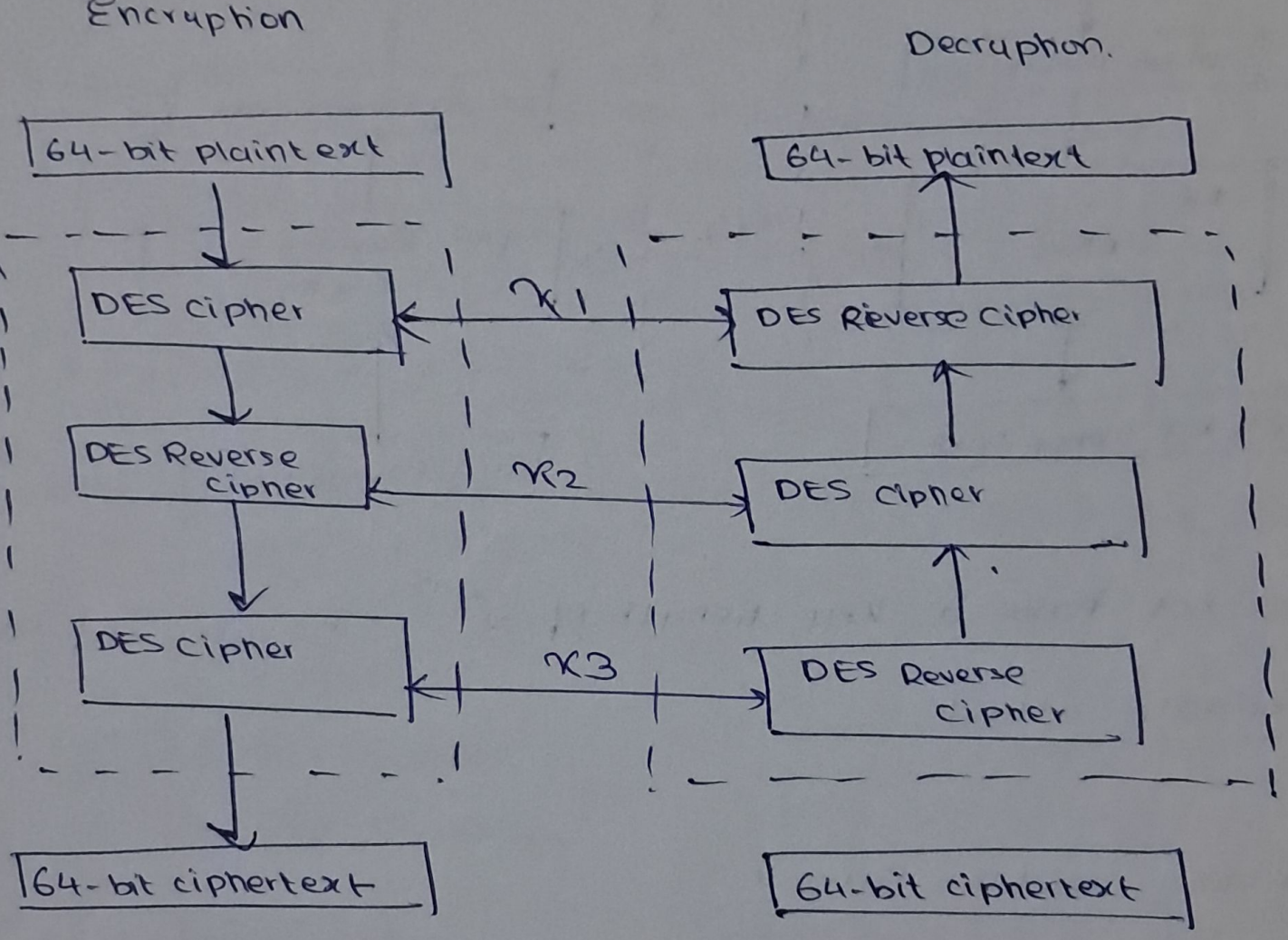
→ Consider that a crypt analyst has a previous pair of P & C, then it can computed all possible values of K_1 and read all values of m.

→ Then for all values of K_2, find m.

→ Compare the m values for X_1 and X_2 and discover a pair which match.

→ The values of X_1 and X_2 where m match are the original key values.

② Triple DES with 3 keys



3TDES key length = 3 × 56 = 168 bits

③ Triple DES with 2 keys

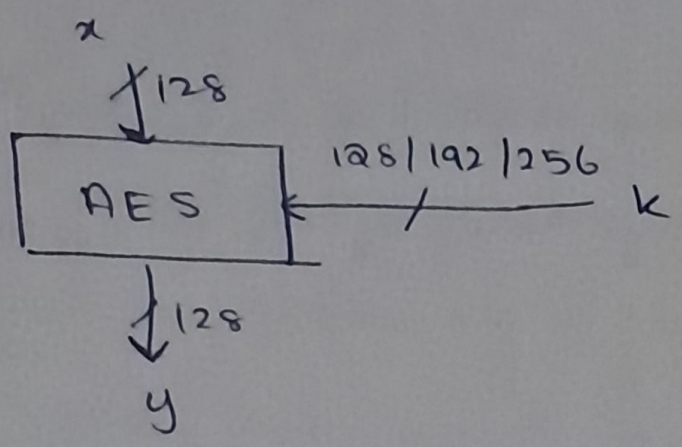
→ same as 3TDES, except that X_3 is replaced by X_1.
That is,

→ Encrypt plaintext blocks w/ X_1, decrypt it with X_2 and finally encrypt with X_1 again

→ 2TDES has a key length of 112 bits.

* Advanced Encryption Standard (AES)

→ most widely used symmetric cipher



→ The number of rounds depends on the chosen key length:

key length (bits)	no. of rounds
128	10
192	12
256	14

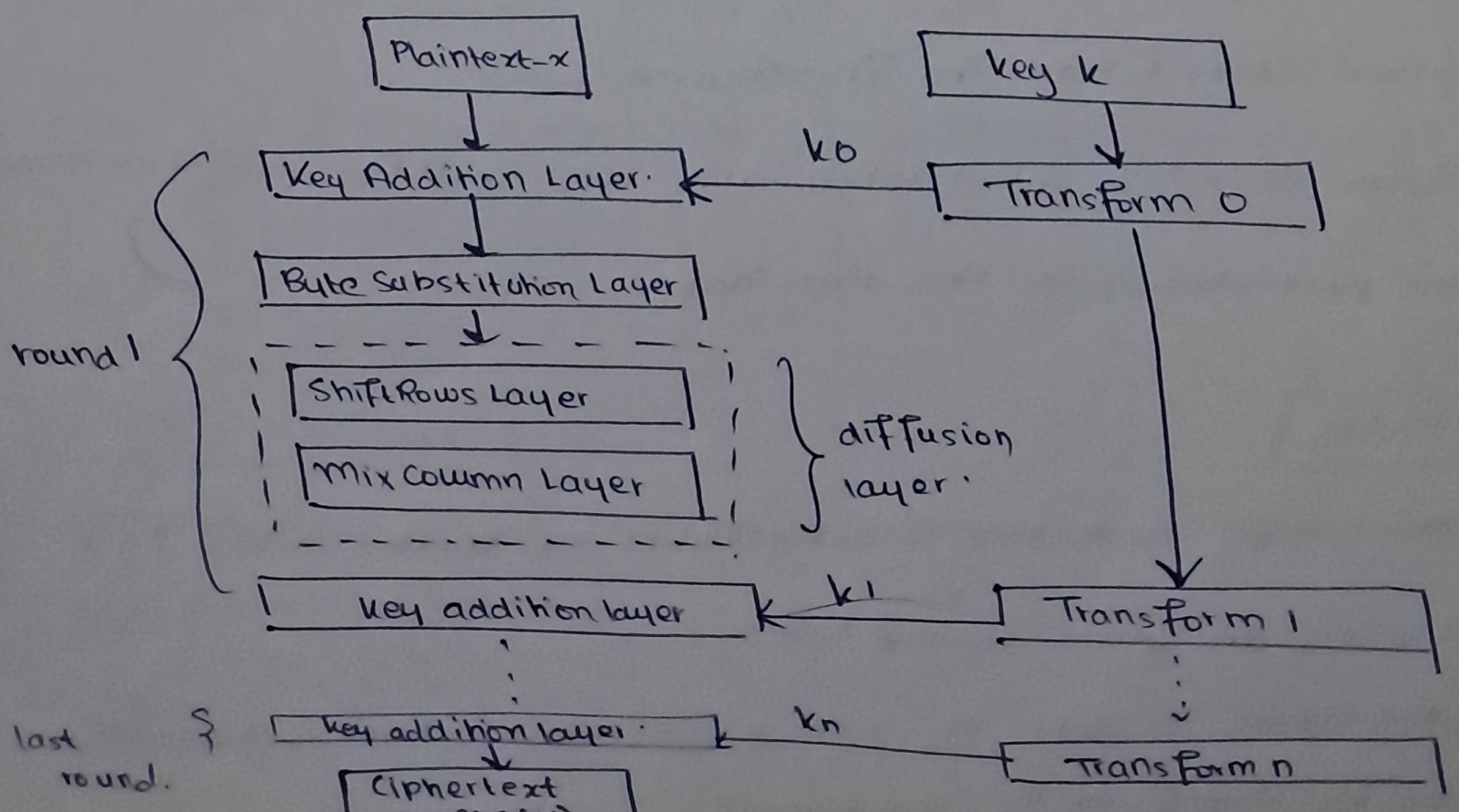
→ The cipher is iterated, with each round consisting of layers. The layers are.

(i) Sub Bytes

(ii) Shift Rows

(iii) mix Columns

(iv) Add Roundkey



Creation of Round Keys

→ A key schedule algorithm is used to calculate all the round keys from the key.

→ The initial key is used to create many different round keys which will be used in the corresponding round of the encryption.

Encryption

→ AES considers each block as a 16 byte grid in the following arrangement:

b0	b4	b8	b12
b1	b5	b9	b13
b2	b6	b10	b14
b3	b7	b11	b15

→ Each round has these 4 steps:

(i) SubBytes

(ii) ShiftRows

(iii) MixColumns

(iv) AddRoundKey

→ The last round doesn't have the MixColumns round.

→ The SubBytes does the substitution and the ShiftRows & MixColumns performs the permutation in the algorithm.

A. SubBytes

→ In this step, each byte is substituted by another byte. It is performed using an S-box lookup table.

→ The substitution is done in such a way that a byte is never substituted by itself or its complement.

→ The result of this step is a 16 byte matrix

B. **ShiftRows**

→ Each row is shifted a particular no. of times.

(i) The first row is not shifted

(ii) The second row is shifted once to the left

(iii) The third row is shifted twice to the left

(iv) The fourth row is shifted thrice to the left

$$\begin{bmatrix} b_0 & b_1 & b_2 & b_3 \\ b_4 & b_5 & b_6 & b_7 \\ b_8 & b_9 & b_{10} & b_{11} \\ b_{12} & b_{13} & b_{14} & b_{15} \end{bmatrix} \longrightarrow \begin{bmatrix} b_0 & b_1 & b_2 & b_3 \\ b_5 & b_6 & b_7 & b_4 \\ b_{10} & b_{11} & b_8 & b_9 \\ b_{15} & b_{12} & b_3 & b_{14} \end{bmatrix}$$

C. **Mix Columns**

→ Each column is multiplied with a specific matrix and thus the position of each byte in the column is changed

→ This step is skipped in the last round

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} & 4 \times 4 & \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

D. **Add Round Keys**

→ The resultant output from the previous stage is XOR'ed with the corresponding round key.

→ This process is repeated for 10/12/14 rounds.

Decryption

→ The order of operations in decryption is as follows:

(i) Add Round Key

(ii) InverseMixColumns ⟶ same as enc. but uses a diff. matrix

(iii) ShiftRows

(iv) InverseSubByte ⟶ Inver S-box is used as a lookup table

* Stream Ciphers and RC4

Stream Ciphers

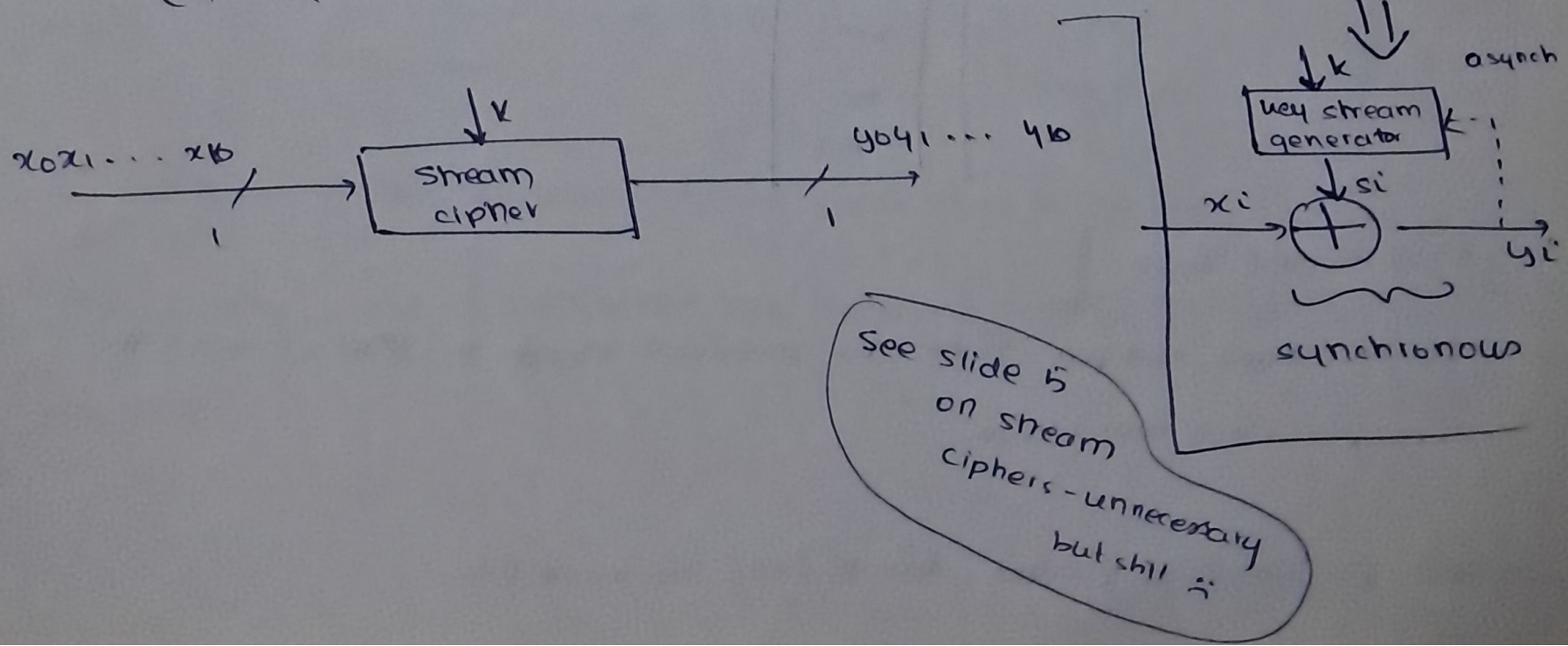
→ A stream cipher is one that encrypts a digital data stream one bit or one byte at a time.

→ This is done by adding a bit from a key stream to a plaintext bit.

→ Stream ciphers are of 2 kinds:

(i) synchronous - key stream depends only on the key

(ii) asynchronous - key stream also depends on the cipher text

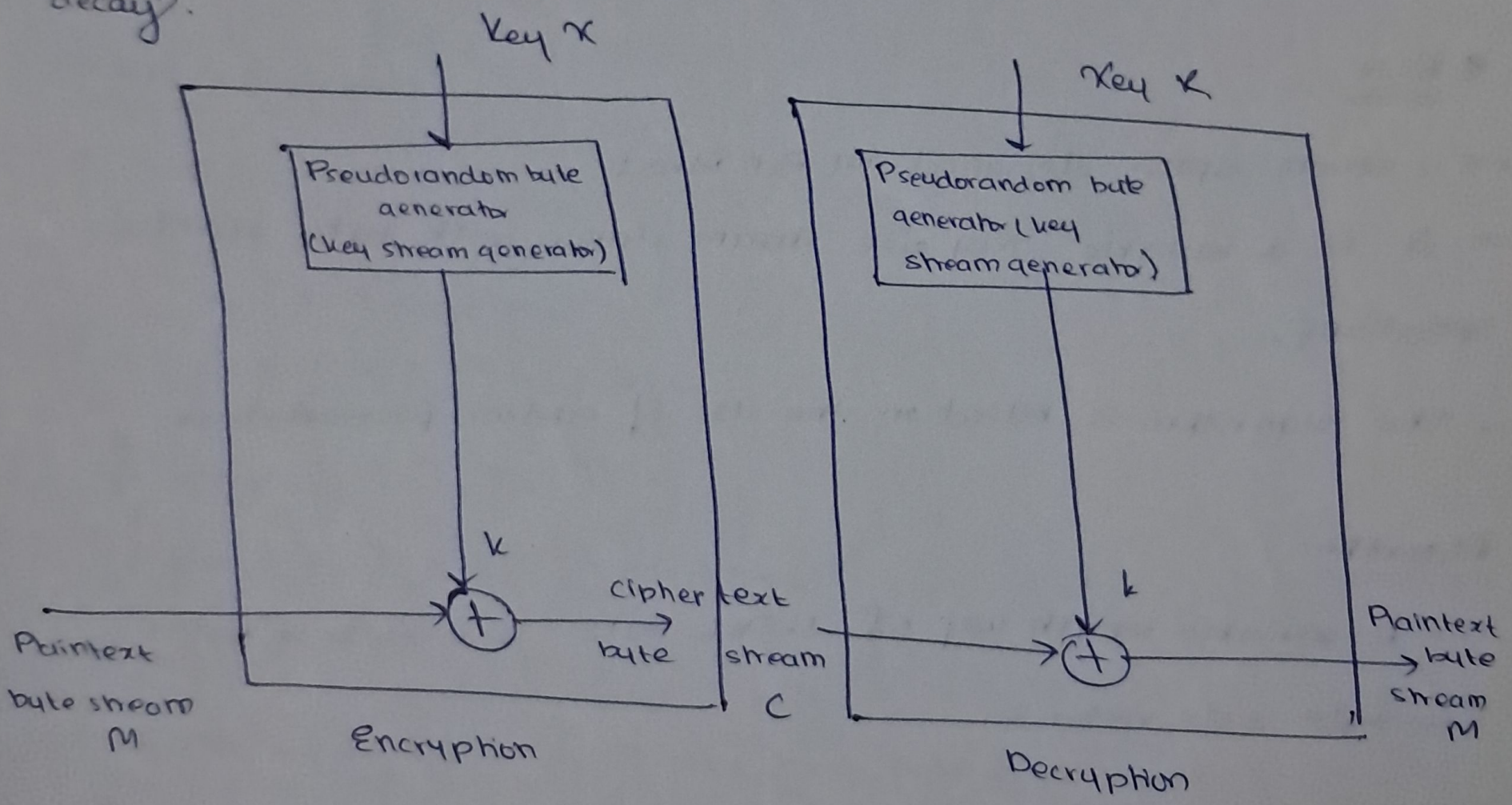


→ An example of a stream cipher is a Vernam cipher.

→ Stream ciphers require the usage of a true random number generator (TRNG)

→ TRNGs are characterized by the fact that their output cannot be reproduced (eg. a flipping a coin 100 times & record the 100 bit sequence)

→ TRNGs are based on physical processes. Examples include coin flipping rolling of dice, semiconductor noise, clock jitter, radio-active decay.



→ The output of the random number generator is called the Keystream, and is combined one byte at a time using the XOR operation

Design Considerations for a Stream Cipher

(i) encryption sequence should have a large period. The longer the period of repeat, the more difficult cryptanalysis becomes.

(ii) The keystream should approximate the properties of a true random generator as much as possible - should have approximately an equal no. of 0s and 1s

(iii) key should be at least 128 bits long.

* RC4

→ a stream cipher designed by Ron Rivest

→ it is a variable key size stream cipher with byte-oriented operations.

→ The algorithm is based on the use of random permutations.

Algorithm

→ A variable length key of 1-256 bytes is used to initialize a 256 byte state vector S

→ At all times, S contains a permutation of all 8-bit nos from 0-255

→ For encryption & decryption, a byte k is generated from S by selecting one of the 255 entries

→ As each value of k is generated, the entries in S are once more permuted.

Initialization

```
for i = 0 to 255 do
    S[i] = i
    T[i] = K[i mod keylen]
```

Initial Permutation

```
j = 0
for i = 0 to 255 do
    j = (j + S[i] + T[i]) mod 256
    swap (S[i], S[j])
```

Stream Generation

```
i, j = 0
while True
    i = (i+1) mod 256
    j = (j + S[i]) mod 256
    swap (S[i], S[j])
    t = (S[i] + S[j]) mod 256
    k = S[t]
```

XOR the value of k with the next byte of plaintext

Strength of RC4

→ exploit biases in keystream to recover repeatedly encrypted plaintexts

→ vulnerabilities found in usage of RC4 in 802.11 LAN networks

* Cipher Block Modes of Operation

Modes of operation : (i) Electronic Code Book (ECB)

(ii) Cipher Block Chaining (CBC)

(iii) Cipher Feedback (CFB)

(iv) Output Feedback (OFB)

(v) Counter (CTR)

① Electronic Code Book

→ simplest mode - each plaintext is handled one block at a time and each block of plaintext is encrypted using the same key.

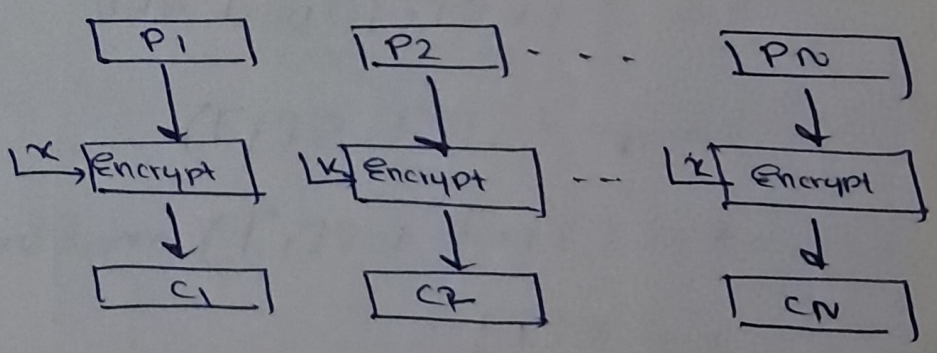
→ codebook ⇒ for a given key, there is a unique cipher text for every b-bit block of plaintext.

→ For a message longer than b bits, break the message into b-bit blocks, padding the last block if needed

∴ ECB :

$$C_j = E(K, P_j)$$

$$P_j = D(K, C_j)$$

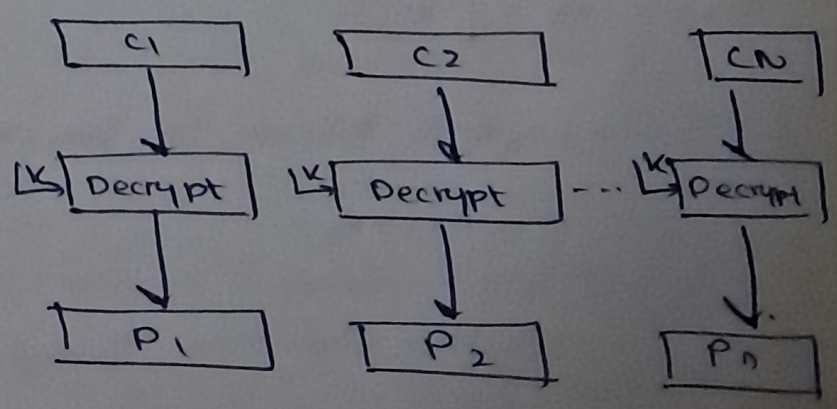


Disadvantages

→ If the same b-bit block appears more than once, it always produces the same cipher text.

→ not secure for longer messages

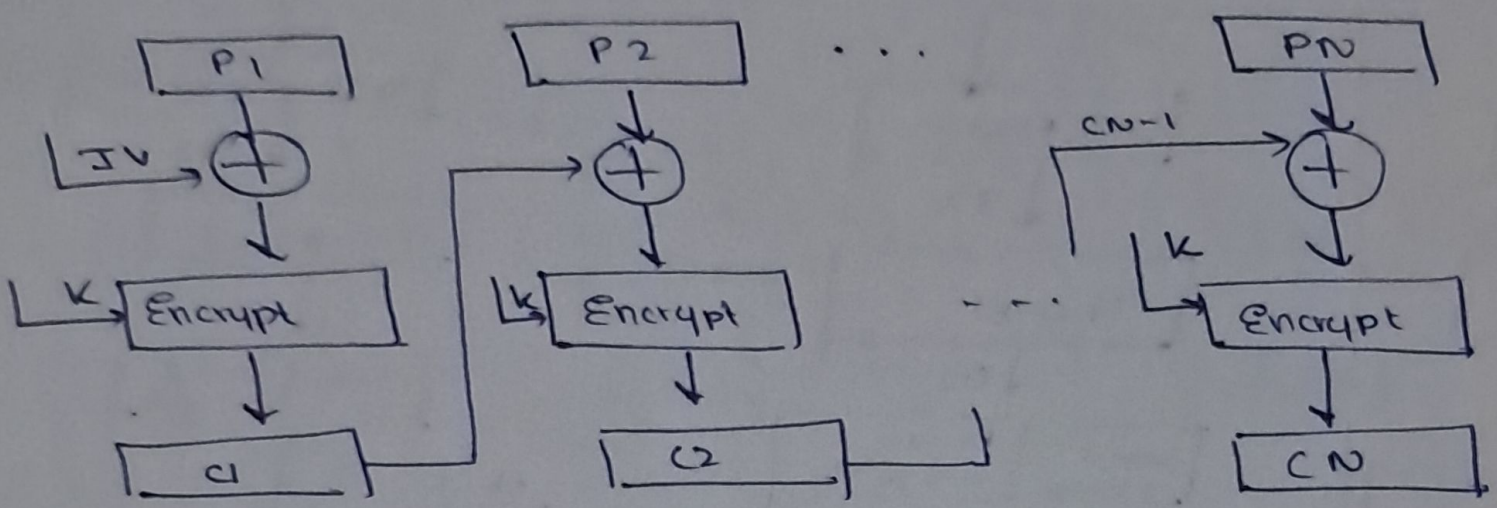
→ cryptanalyst can exploit regularities

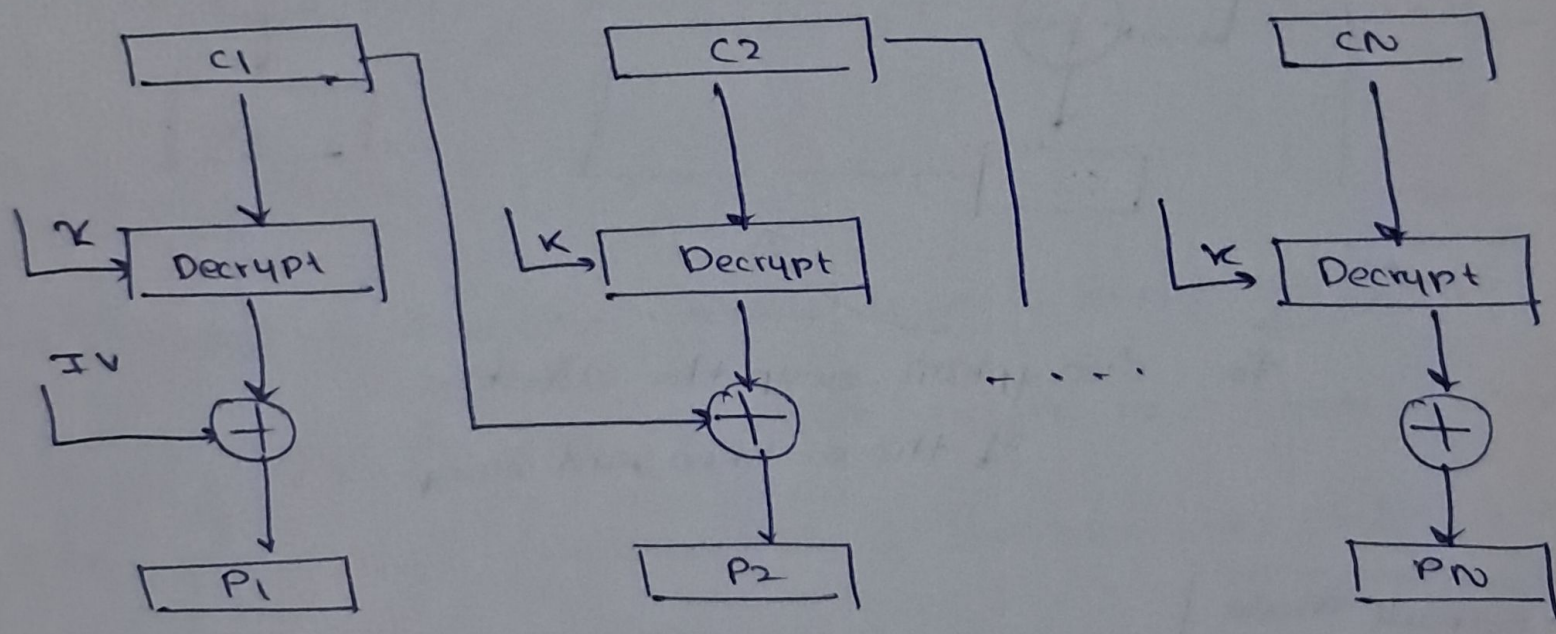


② Cipher Block Chaining Mode (CBC)

→ The input to the encryption algorithm is the XOR of the current plaintext block and the preceding ciphertext block.

→ The input to the encryption function bears no fixed relationship to the plaintext block. Thus, repeating patterns of b are not exposed.





→ CBC is appropriate when the msg. length is greater than b bits

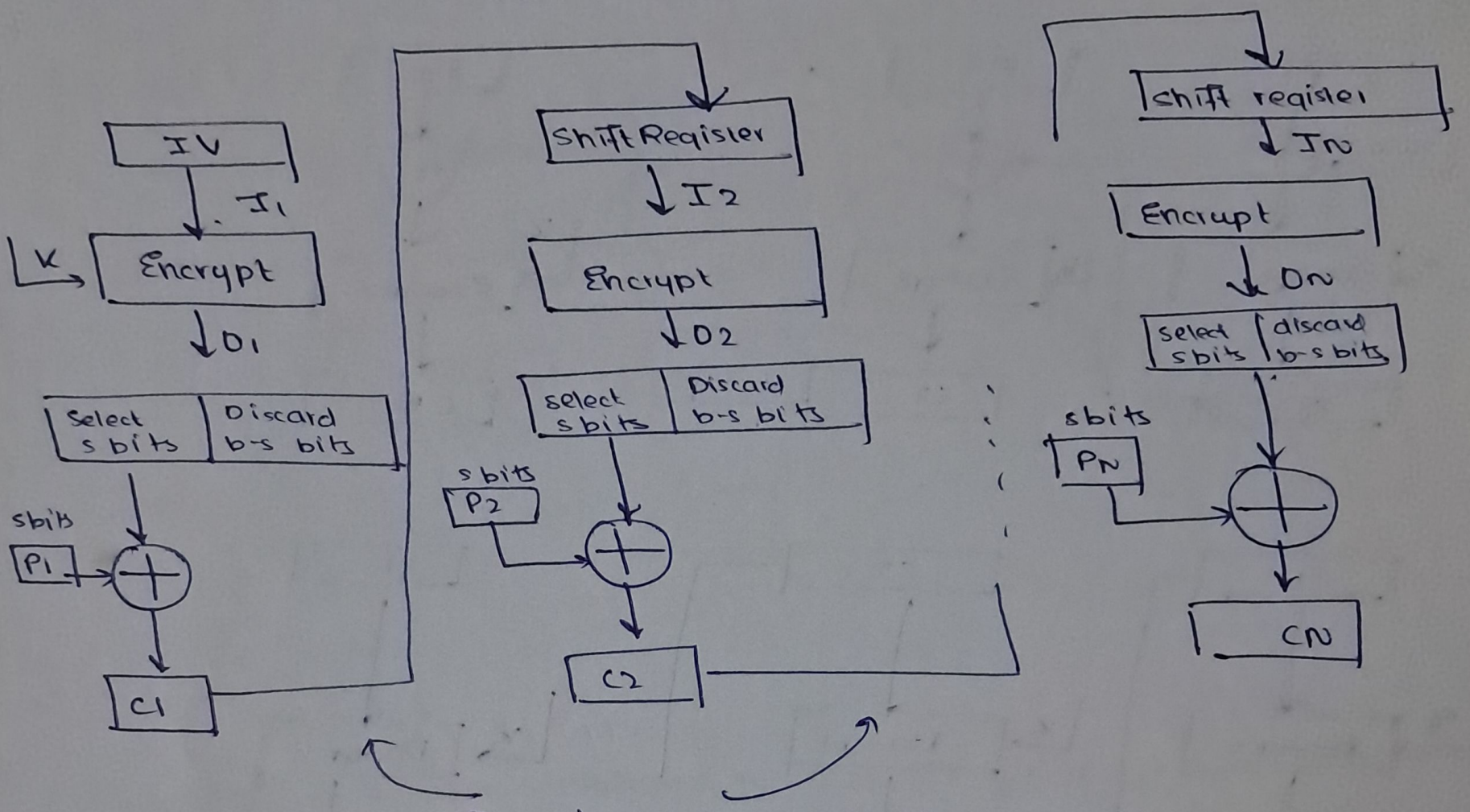
→ CBC can be used for both confidentiality & for authentication.

③ Cipher Feedback Mode (CFC)

→ Like CBC, the units of plaintext are chained together, but rather than using a block of b bits, the plaintext is divided into segments of s bits.

→ The input to the encryption is a b-bit shift register that is set to some initialization vector (IV)

→ The leftmost s bits of the o/p encryption function are XORed w/ the first segment of plaintext to produce the first unit of cipher text C_1, which is then transmitted.



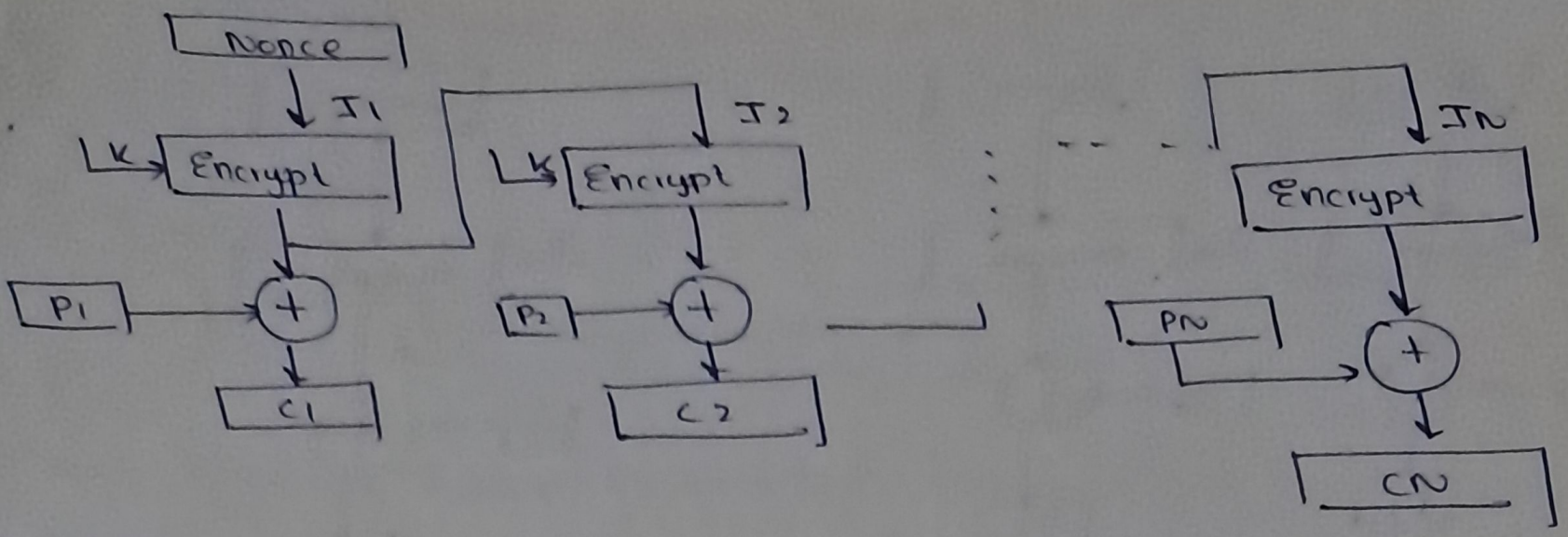
for decryption, swap the direction of these lines, and swap C & P

(4) Output Feedback Mode

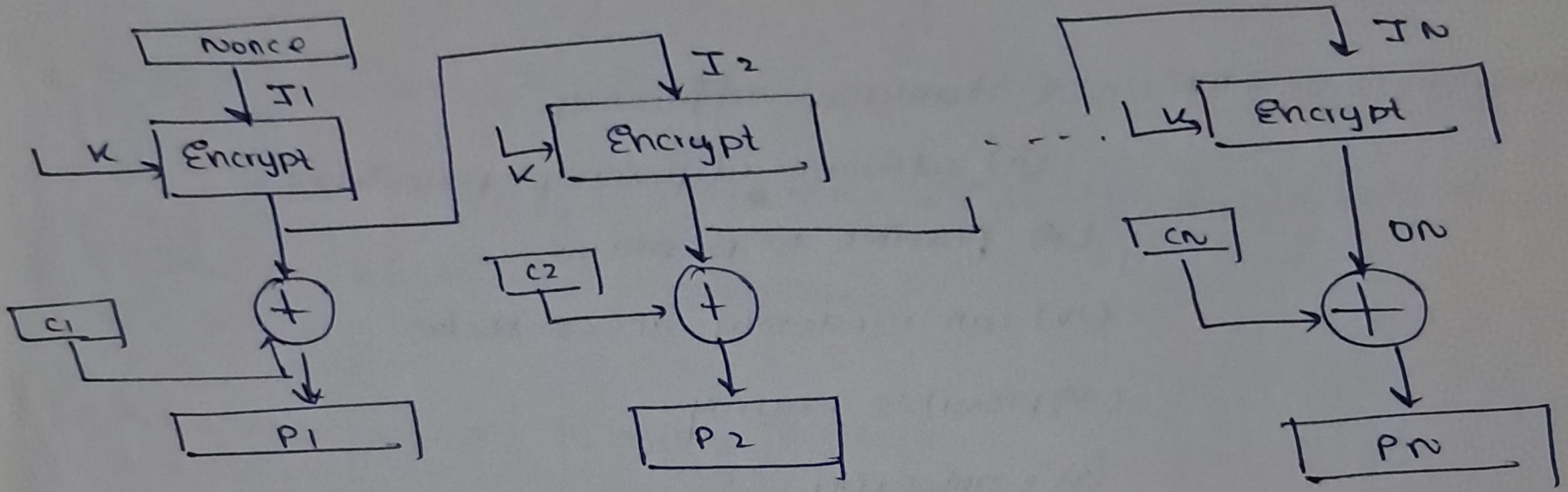
→ Follows the same process as CFB, except that it sends the encrypted output as feedback instead of the actual cipher which is the XOR output.

→ All bits of the block are used rather than s bits

Encryption



Decryption

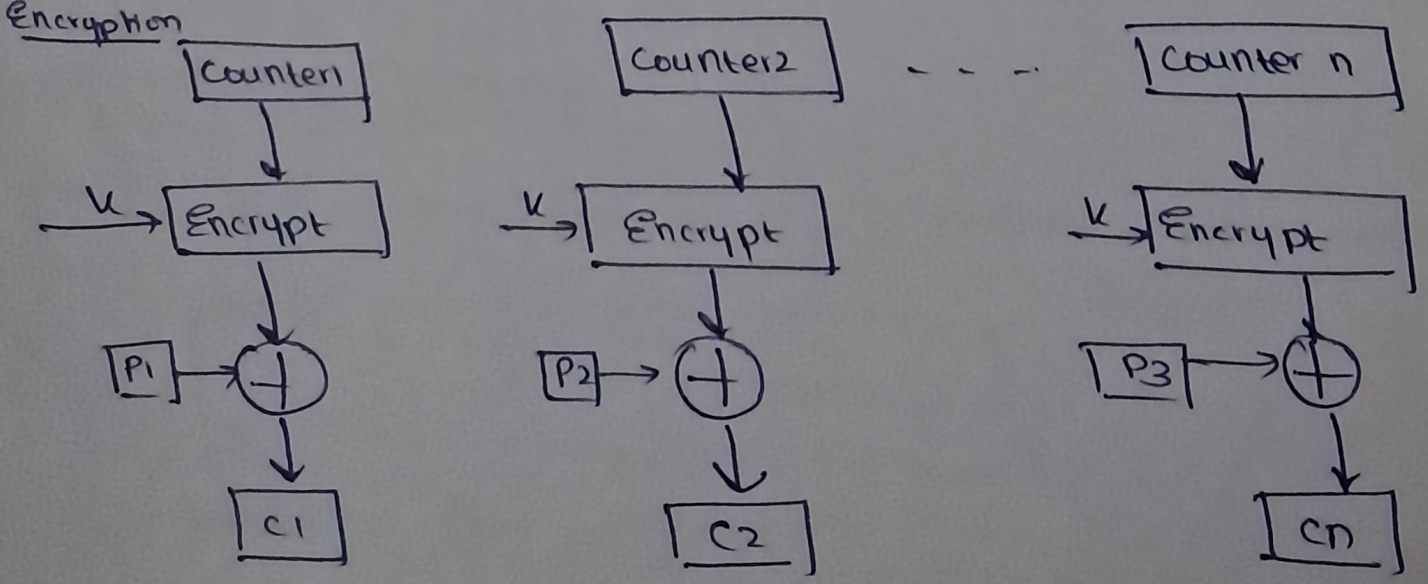


(5) Counter Mode

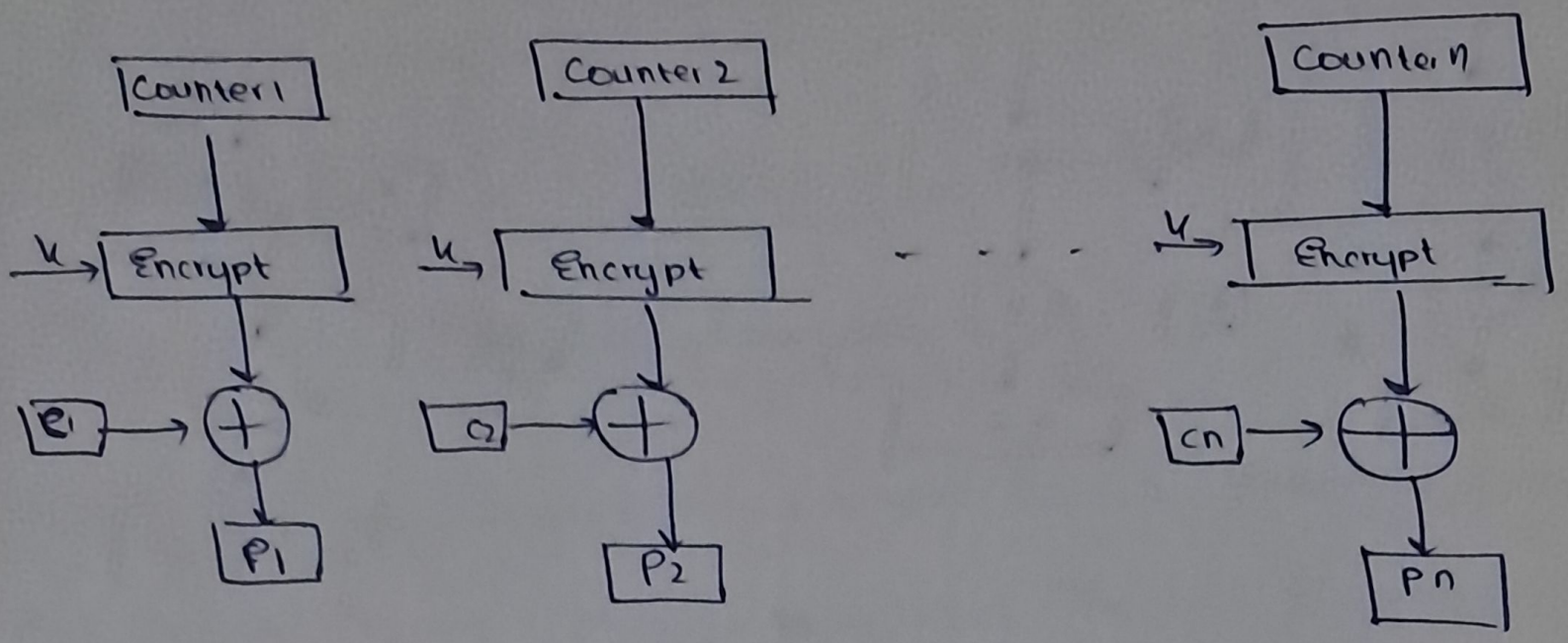
→ Each time a counter - initiated value is encrypted and given as input to XOR w/ plaintext which results in a ciphertext block

→ The CTR mode is independent of feedback use and can thus be implemented in parallel.

Encryption



Decryption



Advantages of CTR - (i) hardware efficiency

(ii) software efficiency - parallelism

(iii) possible to preprocess

(iv) can randomly access blocks

(v) provable security

(vi) simplicity

*Finite Field (Galois Field)