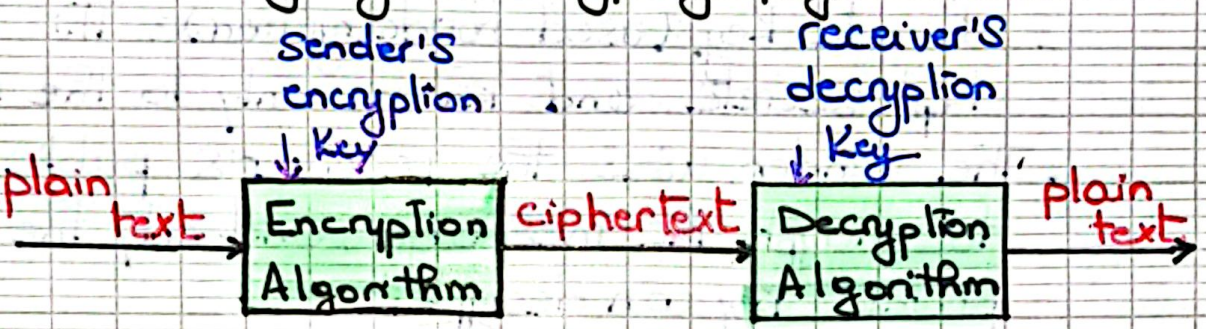
Chapter 4
Symmetric encryption and message confidentiality

1. The language of cryptography



- Definition: Cryptography is the science of securing communication so that only intended recipients can understance the message. It uses mathematical techniques to encrypt (convert plaintext into ciphertext) and decrypt (convert ciphertext back into plaintext).

- Symmetric Key Cryptography
  - In symmetric key cryptography, the same key is used for both encryption and decryption.
  - → The sender and receiver must share the identical secret key in advance.
  - → It's faster and suitable for encrypting large amounts of data.
  - → Challenge: Securely sharing the secret key between the sender and receiver without it being intercepted.
  - → Examples: AES (Advanced Encryption Standard) DES (Data Encryption Standard)

Public-Key Cryptography

→ Public-Key cryptography uses two keys:

1. Public Key (used for encryption): This Key is shared with everyone.
2. Private Key (used for decryption): This Key is Kept Secrete and Known only to the receiver.

→ The sender uses the receiver's public key to encrypt the message, and only the receiver's private key can decrypt it.

→ It eliminates the need to share a secret key but is computationally slower.

→ Examples: RSA, ECC (Elliptic Curve Cryptography)

Comparision

Aspect	Symmetric Key	Public-Key
Keys	Single shared Key	Public and private Key pair
Speed	Faster	Slower
Use Case	Bulk data encryption	Key exchange, digital sign.
Security	Risk in Key distribution	Safer for Key sharing

Cryptography Terms:

1. Plaintext:
   → The original message or data that needs to be protected.
   → This is the readable input to the encryption algorithm

2. Ecryption Algorithm

- A mathematical process that applies substitutions and transformations to the plaintext.
- Converts plaintext into an unreadble format called cipher Text using a secrete Key.
- Examples: AES, RSA or DES algorithms.

3. Secret Key

- A Unique input (like a password or binary value) to the encryption algorithm.
- The transformations performed by the algo. depend on the Key
- Importance: Using different Keys with the same algorithm and plaintext will produce different ciphertexts.

4. Cipher Text

- The encrypted Output, a scrambled, unreadable version of the plaintext.
- Generated by the encryption algo. and secret key.
- Only someone with the correct secret Key can decrypt it back into plaintext.

5. Decryption Algorithm

- Works in the reverse direction of the encryption algorithm
- Takes the ciphertext and the secrete Key as input to reconstruct the original plaintext.

2. Confidentiality Using Symmetric Encryption

Link Encryption

- How it works: Encryption is performed at each communication link, indendently. Data is decrypted and re-encrypted at every intermediate hop (e.g. routers or switches).
- Advantages:
  - Encrypts all data, including headers, routing information, and addresses.
  - User don't need to perform any encryption steps, it operate at a lower OSI layer. (Data Link or Network)
- Disadvantages:
  - Each hop device needs a unique key
  - Packets are decrypted at every hop, exposing data to potential attackers at each point.

End-to-End Encryption

- How it works: Data is encrypted by the sender and remains encrypted until it reaches the receiver. Encryption focuses on the message's payload, leaving headers and routing info untouched for network processing.
- Advantages:
  - Users can control what gets encrypted and how
  - Intermediate devices don't decrypt the data, reducing the risk of exposure.

↳ Disadvantages :
. Does not encrypt headers, addresses or routing info, leaving them susceptible to Traffic analysis

3. Trafic Analysis
. It's the process of monitoring communication flows between parties, which can reveal patterns even if the message content is encrypted

. Challenges:
1. Headers Left Unencrypted (End-to-End)
↳ While the message content is secure, Traffic patterns (e.g. who is communicating, when and how much data) can still be observed.
2. Visible Traffic Volumes
↳ Even with link encryption, network-level Traffic volumes and endpoint activity can expose patterns

. Solution :
Combining end-to-end and link encryption
↳ end-to-end: protects data content over the entire path and supports authentication
↳ Link: Protects headers and mask Traffic flows, reducing monitoring risks.

4. Placement of Encryption in OSI Model.

→ Link Encryption:
Applied at Layer 1 (Physical) or Layer 2 (Data Link)

→ End-to-End Encryption
Applied at higher layers (3, 4, 6, or 7)
- Layer 3 (Network): IPSec
- Layer 4 (Transport): TLS
- Layer 6/7 (Application): HTTPS

→ As encryption moves to higher layers, more entities and keys are involved, making it more secure but also more complex

→ Recap OSI Layers

7 Application
6 Presentation
5 Session
4 Transport
3 Network
2 Data Link
1 Physical

5. Key Distribution

Secure Key distribution is crucial in symmetric encryption, as both parties must share a common secret Key. Many system failures arise from vulnerabilities in Key distribution

→ Key Distribution Methods

1. Direct Key Delivery

- party A selects the Key and physically delivers it to B.
  - Pros: Secure if done correctly
  - Cons: Impractical over long distances or untrusted channels

2. Third-Party Delivery (KDC's)

- A Trusted 3rd party selects the Key and securely delivers it to both A and B.
  - Pros: Avoids direct exchange
  - Cons: Relies on the security of the third party

3. Using an Existing Key

- If A and B have previously shared a Key, it can encrypt a new Key for secure exchange
  - Pros: No need for additional secure delivery
  - Cons: Requires an initial secure Key

4. Relayed by a Third Party

- If both A and B have secure connections to a third party C, C can relay the Key
  - Pros: Secure with a Trusted C
  - Cons: Relies on the availability and Trustworthiness of C.

→ Key Types and Distribution Concepts

1. Session Key:

- A temporary, one-time key used to encrypt all data exchanged during a single communication session.
- Discarded after the session ends, reducing risks if the key is compromised.

2. Permanent Key:

- A long-term key shared between entities, used solely to distribute session keys securely.
- Not used for encrypting user data directly.

3. Key Distribution Center (KDC)

- A trusted entity responsible for managing key distribution.
- Roles:
  - → Authorizes systems to communicate.
  - → Provides a session key for a specific connection upon authorization.
- Centralizes key management, reducing the need for direct key exchanges.

→ Key Distribution Issues

1. KDC Hierachies:
   - ↳ Large networks require multiple KDC
   - ↳ Trust must be established between these KDCs.

2. Session Key Lifetimes
   - ↳ Limiting session key lifetimes enhances security by reducing exposure if a key is compromised.

3. Automatic Key distribution
   - ↳ Systems can distribute key automatically on behalf of users, but users must trust the system's security.

4. Decentralized Distribution
   - ↳ Avoids reliance on a central KDC. but requires robust protocols for trust and security

5. Key Usage Control
   - ↳ Keys should be restricted to specific purpose (e.g. encryption, authentication) to minimize misuse.

6. Notes about Data Security.

→ Physical Security of Devices.

- Equipment and devices have to be locked away securely, with proper access control in place
- Different ways of physical security of devices:
  - ↳ Log Equipment: Tools or devices used for recording, monitoring, or analyzing data activities in a system
  - ↳ Cable Locks: Security devices consisting of a flexible cable and lock mechanism used to secure physical devices.
  - ↳ Access control: Security mechanism that regulates who or what can view, use, or modify resources in a system, based on predefined permissions.

→ Backup Procedure

- Backup procedure is essential, for both business and personal users.
- We should schedule backup frequently
- Incremental backup: Backing up files that have been modified since the last full backup.
- Methods of backups:
  - ↳ Back-up to a device
  - ↳ Remote Backup Service (AKA Cloud Backup)

→ Permanently Deleting Data

- Deleting data from devices or drives means to remove it permanently
- 2 main reasons for deleting data:
  - ↳ Save Space
  - ↳ Security
- Deleting and permanently destroying data:
  - ↳ When deleting data it's moved to Recycle Bin, once the Recycle Bin is emplied, the files are permanently deleted.
- Common Methods of Permanently Destroying Data:
  - ↳ Shredding: Destroying materials into small pieces to ensure secure disposal of sensitive info.
  - ↳ Drive/Media destruction: Physically destroying storage devices (hard drives, USBs) to prevent data recovery
  - ↳ Degaussing: Using a strong magnetic field to disrupt and erase data stored on magnetic media.
  - ↳ Data Destruction Utilities: Software tools designed to securly erase data from storage devices.

→ Protect Ourself When Online

It's important to ensure that all online purchases and financial transactions should be taken on secure websites.

↳ Secure Websites

- Set to prevent unauthorized people from seeing info. that is sent to or from those sites
- Secure website can be identified by:
  - ↳ The URL : https
  - ↳ A lock icon displayed on the status bar

→ Phaming

Scamming concept in which malicious code is installed on a computer or server through hacking, misdirecting users to fraudulent and fake websites.

→ Digital Certificates

An electronic identity card that establishes your credentials when doing business or other transactions on the web.

→ One-Time Password (OTP)

- Unique and time sensitive pass. used as added security to sensitive info.
- Valid for only one login session or transaction

- must be used within a specific time period
- doesn't replace the original username and password
- it only provide a second layer of security
- The pass. will be sent via SMS or email

→ COOKIES

- Term used to describe a type of message that is given to a web browser by a web server
- The main purpose of it is to identify users and possibly prepare customized web pages or to save site login info.
- Types of cookies:
  - ↳ Session cookie: stored in Temporary memory and is erased when the user close the browser
  - ↳ Permanent cookie: stored on the hard drive until expiry or until deleted

→ Content - Control Software

- Censorware or Content - Blocking software
- Software designed for controlling what content is allowed to be viewed by a user and the amount of time to be spent using the internet and computer activities
  - ↳ Internet Filtering Software: Limit or restrict what a user can view on the internet
  - ↳ Parental Control Software: restrict children as to what can be viewed on net, time spent on the comp. and what files can be accessed.

7. Protocols for Secure Communications

→ 4 Steps to ensure Security

1. Confidentiality : Only sender, intended receiver should "understand" message contents
   ↳ Sender encrypts message
   ↳ receiver decrypts message
2. Authentication : Sender, receiver want to confirm identity of each other.
3. Message Integrity : Sender, receiver want to ensure message not altered (in transit, or afterwards) without detection.
4. Access and Availability : Services must be accessible and available to users.

→ Securing Internet Communication with SSL and HTTPS

SSL (Secure Sockets Layer) and its successor TLS (Transport Layer Security) are protocols used to secure internet communication by encrypting data exchanged between a client (e.g. a web browser) and a server.

↳ SSL/TLS ensures Confidentiality, Integrity, and Authentication by encrypting data, preventing interception or tampering. (use public key encryption)

↳ HTTPS is the secure version of HTTP which uses SSL/TLS to encrypt communication between the client and the server.

→ Securing Web Transactions with SET, SSL and, HTTPS

. Secure Electronic Transactions (SET)
  ↳ Developed by MasterCard and VISA to provide protection from electronic payment fraud
  ↳ Uses DES to encrypt credit card info. transfers and RSA for Key exchange
  ↳ Provides security for both Internet-based credit card transactions and credit card swipe systems in retail store

. SET, SSL, HTTPS, Secure Shell (SSH-2) and IP Security (IPSec) work to secure web browsers, espacially at electronic commerce sites

→ Securing Email with S/MIME, PEM and PGP

. S/MIME (Secure/Multipurpos Internet Mail Extensions): A standard for public key encryption and digital signitures for email. It provides confidentiality by encrypting the email content and authentication by verifying the Sender's identity to ensure confidentiality and integ.

. PEM (Privacy-Enhanced Mail): A format for encoding cryptographic data, including public and private keys, certificates and email messages. It supports email encryption and Signing, Uses 3DES symmetric key encry.

Use RSA for sym. Key exchange.

. PGP (pretty Good Privacy): Popular encryption standard for securing emails. It uses a combination

of Symmetric and asSymmetric encryption to ensure confidentiality, authenticity and integrity of emails message. Use IDEA Cipher for encoding PGP, PEM and S/MIME works to secure email operations.

→ Securing Wireless Networks with WEP and WPA.

- Wired Equivalent Privacy (WEP): Early attempt to provide security with the 802.11 network protocol.
- Wifi Protected Access (WPA and WPA2): Created to resolve issues with WEP
- Next Generation Wireless Protocols: Robust Secure Networks (RSN).

AES-Counter Mode Encapsulation
AES-OFFset Codebook Encapsulation

- Bluetooth: Can be exploited by anyone within approximately 30 foot range, unless suitable security controls are implemented.

→ Securing TCP/IP with IPSec.

- Internet Protocol Security (IPSec)
- Open Source protocol to secure communications across any IP-based network.
- IPSec designed to protect data integrity, user confidentiality, and authenticity at IP packet level.

- IPsec combine different cryptosystems:
  - Diffie-Hellman Key exchange, Public Key Cryp.
  - Bulk encry. algo., Digital Certificates
- In IPSec IP layer security obtained by use of application header (AH) protocol or encapsulating security payload (ESP) protocol.
  - Defines the info. to add to an IP packet, as well as how to encrypt packet data.

8. Symmetric Encryption Tools

1. Substitution: Replaces plaintext characters with ciphertext characters based on a defined rule / Key.
2. Transposition: Rearrange the order of characters in plaintext to produce ciphertext
3. Exclusive OR (XOR): Combines plaintext and key bits using XOR logical operation.
4. Logical Shift: Shifts bits of plaintext. or right based on the key
5. Combination: A mix of the above methods for added security.

8.1. Substitution Cipher

→ **Monoalphabetic Substitution Cipher:**

- Each plaintext character is replaced with a corresponding ciphertext character based on a predefined mapping or function
- Key: A table or rule defining charac. mappings

- Example 1:
  - → Keyword : ANDREW DICKSON WHITE
  - → Cipher Alphabet:

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

- Example 2:
  - → Encryption Rule:

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

- → Plaintext : have a nice weekend
- → Ciphertext: GXEHXSZYHKHWHSZ

- → Decryption: The decryption rule reverses the mapping

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

- This restores the original plaintext.

- → Increment Substitution Cipher (Caesar Cipher)
  - plaintext : VOYAGER
  - Key : +3 (each letter is shifted forward by 3 positions in the alphabet)
  - Ciphertext : YRBDJHU
  - The Decryption rule will be the reverse (-3)

→ Complex Substitution Cipher Using a Patterned Key

- plainText : VOYAGER
- Key : "123" => +1, +2, +3, +1, +2, +3, +1
- Ciphertext: WQABBIHS

↳ Each character in the plaintext is increment by the corresponding value in the Key.

→ Simple Polyalphabetic Substitution :

a) Vigenère Cipher

- The vigenère cipher uses multiple caesar ciphers based on a repeating Keyword. Each letter in the plaintext is shifted by a value determine by the corresponding Key letter in the :
- Example:
  - plainText : HELLO EVERYONE
  - Key = CDBE (repeated as needed)
  - The Key is composed of 4 letters, then we divide the plaintext into blocks of 4 letters M1 : HELL, M2 : OEVE, M3 : RYON, M4 : E
  - Shift each letter in the block by the corresponding Key letter's value (e.g. C = +2, D = +3, B = +1, E = +4)
  - Ciphertext : JHMPQHWITBPRG

b) One-Time Pad (OTP)

- The OTP encrypts each character of the plaintext by combining it with a truly random key of the same length, using modular addition
- Characteristics:
  - Proven unbreakable if the key is random, as long as the plaintext, and never reused.
  - Difficult to implement due to the need for large secure, random keys.
- Example:
  - Plaintext: HELLO
  - Key: XMCKL
  - Ecrypt each character:

H + X = E
E + M = Q
L + C = N
L + K = V
O + L = Z

} Ciphertext: EQNVZ

8.2. Transposition Cipher:

→ Simple Transposition

In a simple transposition, the characters of the plaintext are rearranged based on a key's positional order.

- Example:
  - Plaintext: HELLO EVERYONE.
  - Key: CBVA (size = 4)
  - alphabetical order 3241

→ Divide plaintext into blocks of size 4:
HELL OEVE RYON E
→ Rearrange based on the Key's order (3 → 2 → 4 → 1)
→ Ciphertext: LELH VEEO OYNR E

→ Columnar Transposition
. In a columnar transposition, the plaintext is written in rows, and the columns are reordered based on a Keyword.
. Example:
→ plaintext: WE ARE DISCOVERED FLEE AT ONCE
→ Keyword: ZEBRAS (size = 6)
→ alphabetical order: A B E R S Z (1 2 3 4 5 6) ⇒ 632415

→ Write the plaintext as rows

6	3	2	4	1	5
W	E	A	R	E	D
I	S	C	O	V	E
R	E	D	F	L	E
E	A	T	O	N	C
E	X	X	X	X	X

→ Then we read the cipher text by column in order (1, 2, 3, ..., 6)
Ciphertext: EVLNACDTXESEAXROFOXDEECWIREE

→ Decryption:

EVLNXACDTXESEAXROFOXDEECXWIREE

→ Write the cipher text as columns

Size of the matrix: length(ciphertext)/length(Key)

⇒ 30/6 = 5

6	3	2	4	1	5
W	E	A	R	E	D
I	S	C	O	V	E
R	E	D	F	L	E
E	A	T	O	N	C
E	X	X	X	X	X

→ Then we read the plain text by row

WE ARE DISCOVERED FLEE AT ONCE

9. Block Cipher Modes of Operation

. Block cipher modes are methods to encrypt and decrypt data in fixed-size blocks. Each mode has unique features and use cases.

1) ECB (Electronic Codebook Mode)

. Process:

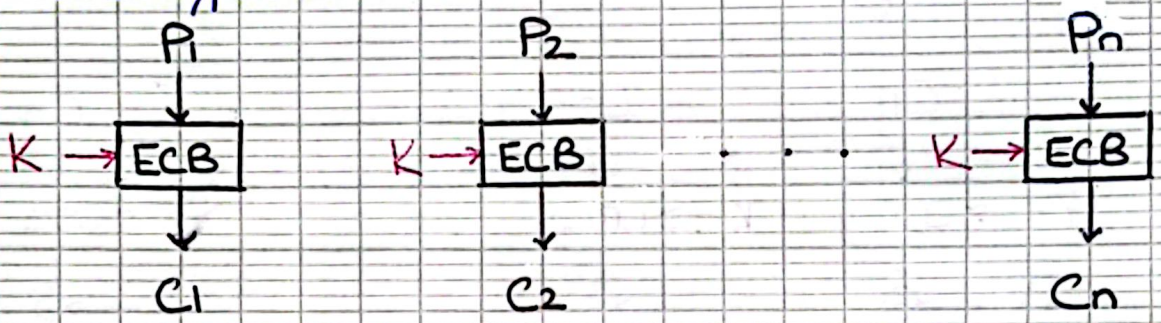
→ Message is broken into independent blocks.

→ Each block is encrypted idependently using the same Key.

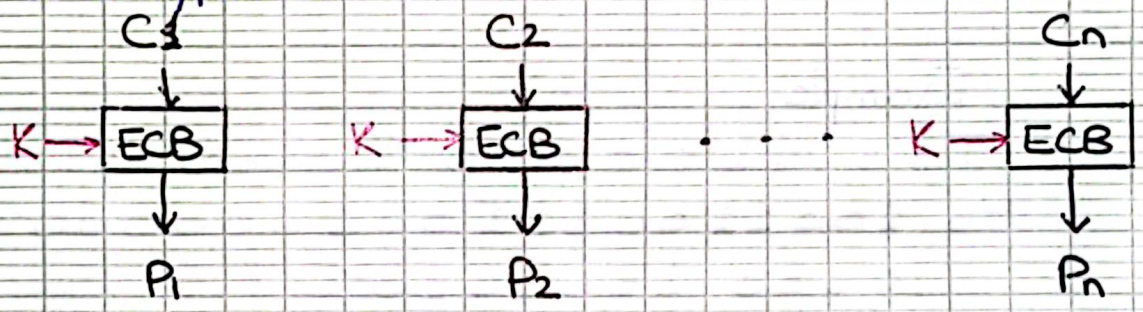
$C_i = K(P_i)$

C_i → Cipher Text, K → Key, P_i → plain Text

- Encryption :



- Decryption :

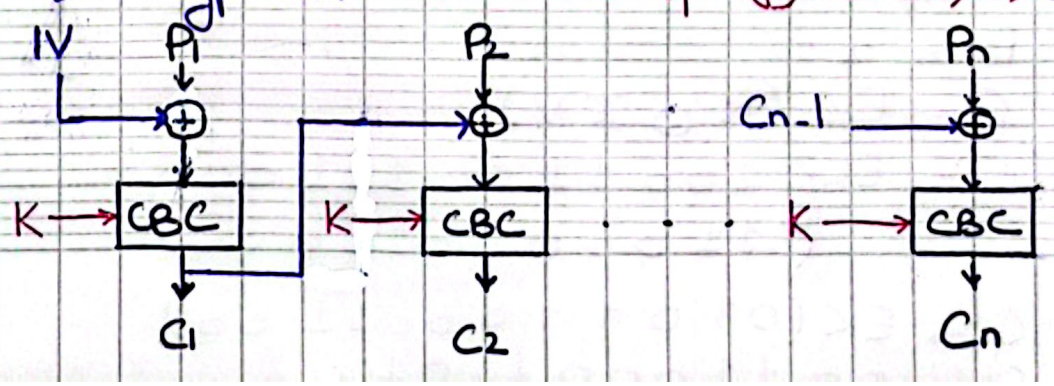


2) CBC (Cipher Block Chaining Mode)

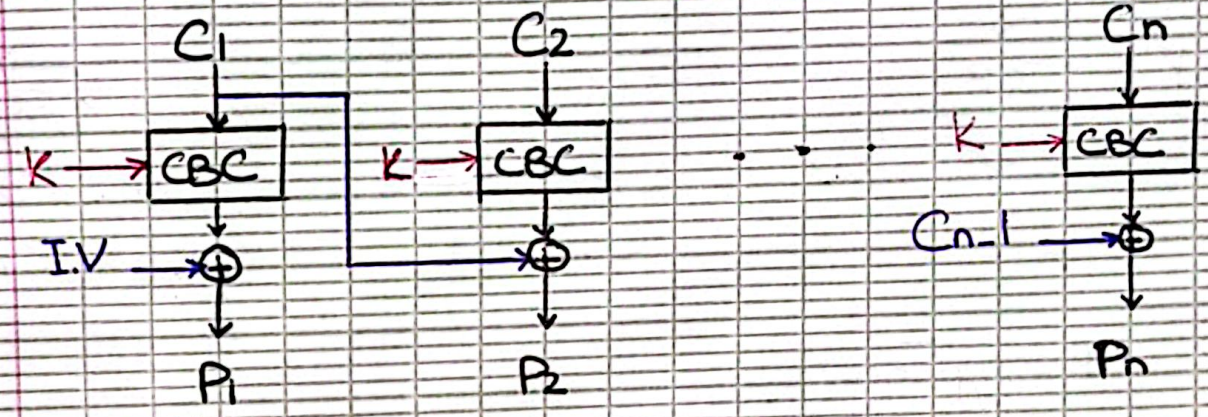
- Process :
  - ↳ Message is broken into blocks
  - ↳ These blocks are linked together in the encryption operation
  - ↳ Each plaintext block is XOR ed with the previous cipher text block before encryption
  - ↳ Requires an Initialization Vector (IV) for the first block

XOR ⊕ :
00 ⇒ 0
01 ⇒ 1
11 ⇒ 0
10 ⇒ 1

- Encryption : $C_i = E_K (p_i \oplus C_{i-1}) \Rightarrow C_0 = IV$



. Decryption: $P_i = D_K(C_i) \oplus C_{i-1} \Rightarrow C_0 = IV$



. Example:

I.V = 11 0 0 0 0 0 0 1 0 0 0 0 (Size = 12)

P = 1 0 0 1 1 0 0 1 1 1 1 0 0 | 1 0 0 0 0 0 0 0 1 1 0 1 0
0 0 0 0 1 0 0 1 0 1 0 1

K: "Reverse the order of each block"

. Encryption:

1) Decompose the plain text into blocks based on the I.V size (12)
2) Pad with extra bits (e.g 0 or 1)
3) For each block of plaintext:
   - xor the block with the previous ciphertext (IV for the first block)
   - Encrypt the result using the provided key
   - Output the encrypted block as the ciphertext for the current block.

$C_1 = E_K(p_1 \oplus I.V)$

$C_1 = E_K(100\,11\,00\,111\,00 \oplus 11\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0)$

$C_1 = E_K(01\,0\,1\,1\,0\,0\,0\,1\,1\,00)$

$C_1 = 00\,11\,0\,0\,0\,11\,0\,10$

$C_2 = E_K(P_2 \oplus C_1)$

$= E_K(1\,0\,0\,0\,0\,0\,0\,0\,1\,1\,0\,1\,0 \oplus 0\,0\,1\,1\,0\,0\,0\,1\,1\,0\,1\,0)$

$= E_K(1\,0\,1\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0)$

$= 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,0\,1$

$C_3 = E_K(P_3 \oplus C_2)$

$= E_K(0\,0\,0\,0\,0\,1\,0\,0\,1\,0\,1\,0\,1 \oplus 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,0\,1)$

$= E_K(0\,0\,0\,0\,0\,1\,0\,0\,1\,1\,0\,0\,0)$

$= 0\,0\,0\,0\,1\,1\,0\,0\,1\,0\,0\,0\,0$

$C = C_1\,C_2\,C_3$

$= 0\,0\,1\,1\,0\,0\,0\,1\,1\,0\,1\,0\;0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,0\,1$
$0\,0\,0\,0\,1\,1\,0\,0\,1\,0\,0\,0\,0$

. Decryption ;

$P_1 = D_K(C_1) \oplus I.V$

$= 0\,1\,0\,1\,1\,0\,0\,0\,1\,1\,0\,0$
$\oplus 1\,1\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0$
$= 1\,0\,0\,1\,1\,0\,0\,1\,1\,1\,0\,0$

$P_2 = D_K(C_2) \oplus C_1$

$= 1\,0\,1\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0$
$\oplus 0\,0\,1\,1\,0\,0\,0\,1\,1\,0\,1\,0$
$= 1\,0\,0\,0\,0\,0\,0\,0\,1\,1\,0\,1\,0$

$P_3 = D_K(C_3) \oplus C_2$

$= 0000\,1001\,1000$

$\oplus\ 0000\,0000\,1101$

$= 0000\,1001\,0101$

$\Rightarrow P = 1001\,1001\,1110\,0100\,0000\,0011\,0101\,0000$
$1001\,0110$

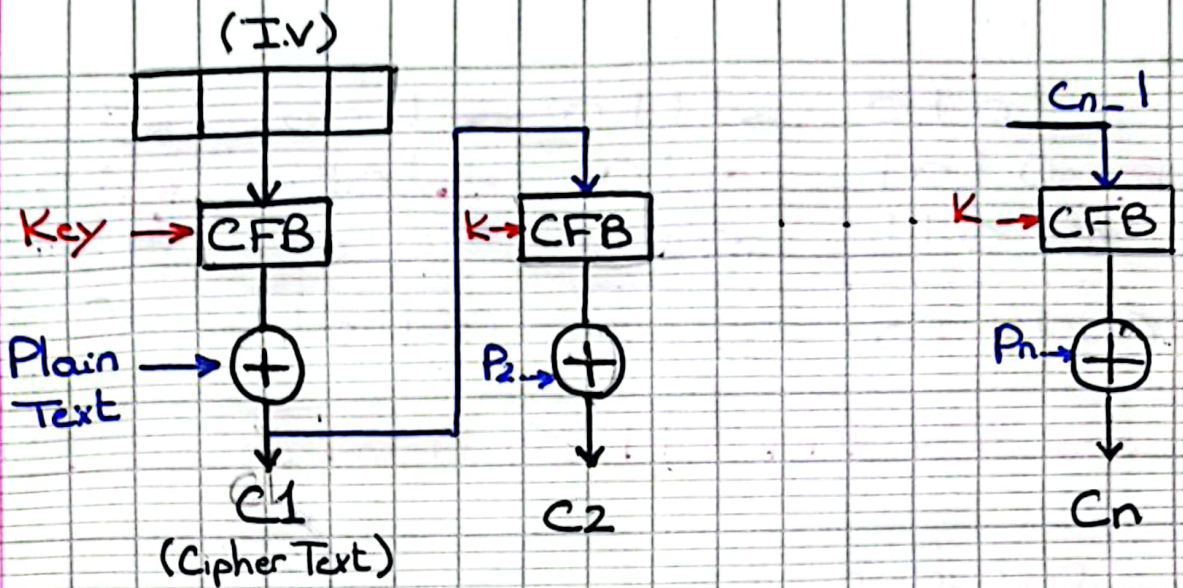
3) Cipher Feed Back (CFB)

- Process
  - Message is treated as a stream of bits (small chuncks of data) instead of entire blocks.
  - We start with an Initialization Vector (IV)
  - Encrypt the IV using the block cipher
  - The encrypted IV is XORed with the first chunk of plaintext to create the first piece of ciphertext
  - The ciphertext produced replaces the IV
  - This new value is encrypted again and XORed withe next plaintext chunk
  - This process continues for the entire message

- Encryption:

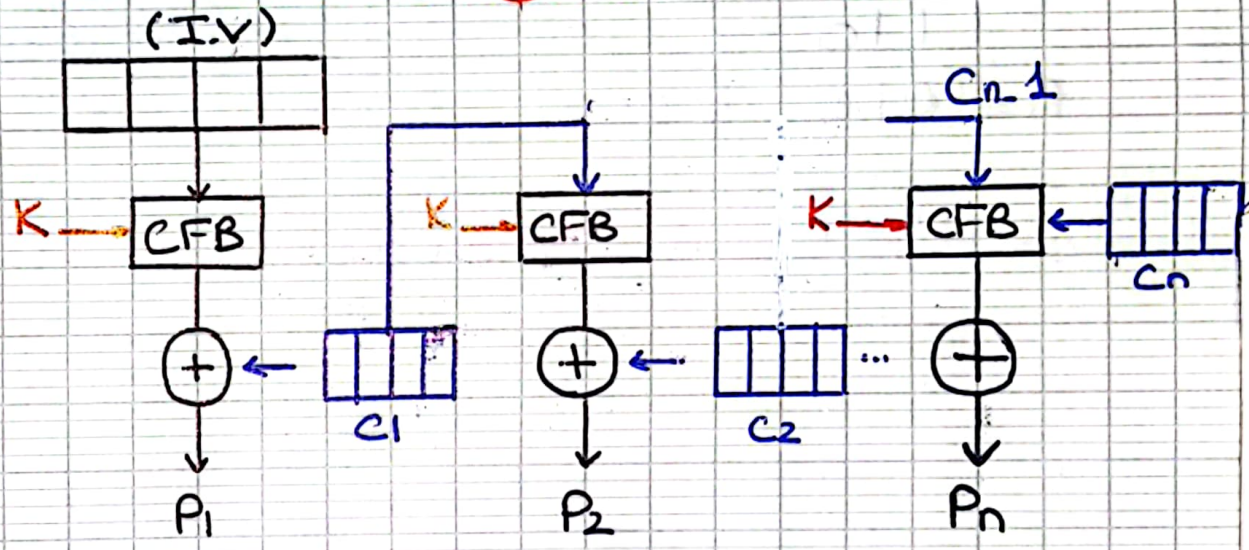
$C_i = E_K(c_i - 1) \oplus P_i$

$C_0 = I.V$



. Decryption:

$P_i = E_K(C_{i-1}) \oplus C_i$ $\quad C_0 = I.V$



. Example:

I.V = 1010

M = 1011/0001/0100/101

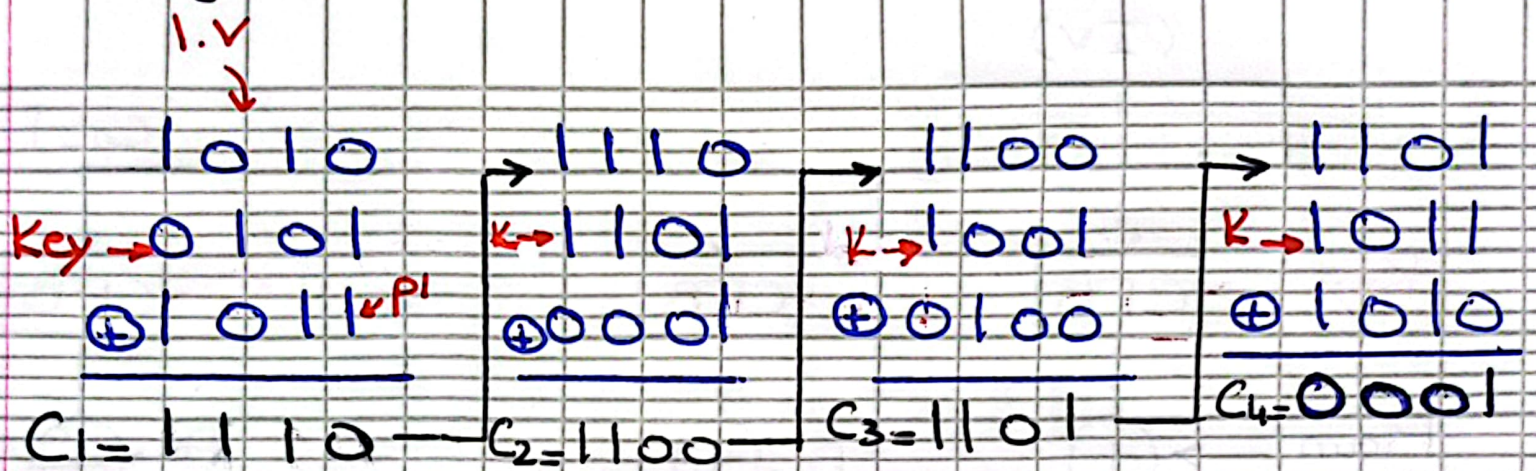
Key: 1 → 2

2 → 3

3 → 4

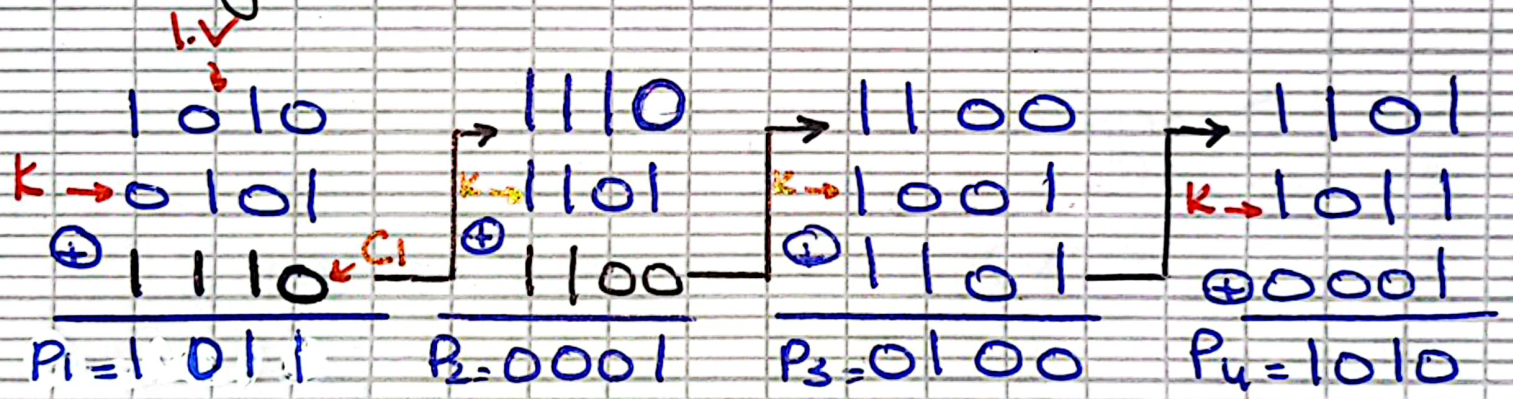
4 → 1

· Encryption :



<=> C = 1110 1100 1101 0001 *

· Decryption :



<=> P = 1011 0001 0100 1010