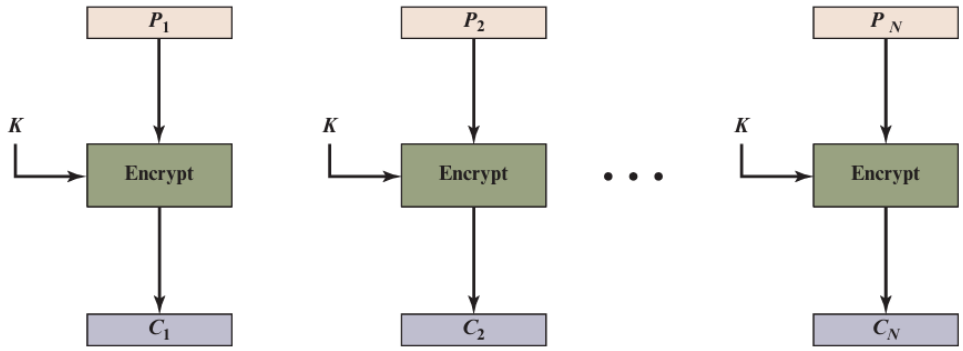
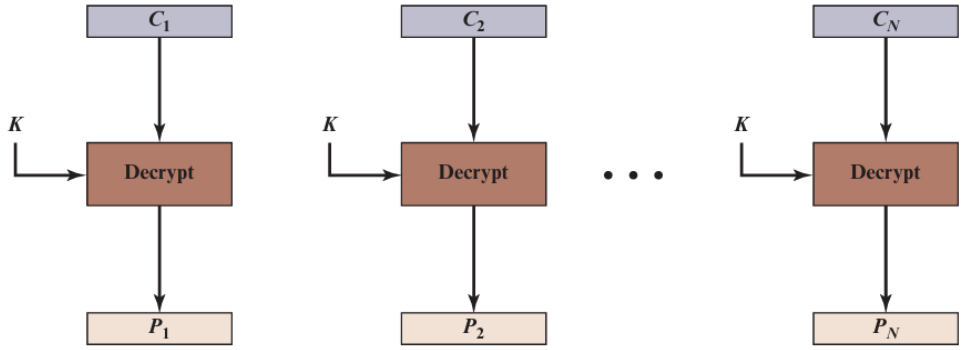
Course: Data Security
Lecturer: Dr. Mahmoud Yehia
TA: Eng. Eslam Osama
Block Cipher Operation Modes

(1) Electronic Codebook (ECB)

Each block of plaintext bits is encoded independently using the same key.



(a) Encryption



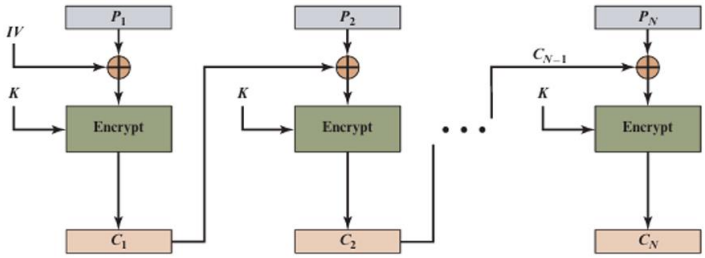
(b) Decryption

ECB	$C_j = E(K, P_j) \quad j = 1, \ldots, N$	$P_j = D(K, C_j) \quad j = 1, \ldots, N$

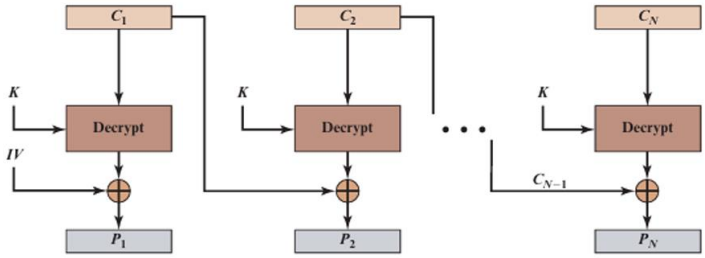
Advantages	Disadvantages
Can be parallelized — each block can be encrypted/decrypted independently.	Not suitable for large data as it may reveal repetition — patterns in plaintext remain visible, leaking structure. (Less secured)
Errors in one block do not propagate to others.	Plaintext must be a multiple of block size — requires padding for short or misaligned messages.
	Requires both encryption and decryption functions — increases software and hardware complexity.

(2) Cipher Block Chaining Mode (CBC)

The input to the encryption algorithm is the XOR of the next block of plaintext and the previous block of ciphertext.



(a) Encryption



(b) Decryption

CBC	$C_1 = \mathrm{E}(K, [P_1 \oplus \mathrm{IV}])$ $C_j = \mathrm{E}(K, [P_j \oplus C_{j-1}]) j = 2, \ldots, N$	$P_1 = \mathrm{D}(K, C_1) \oplus \mathrm{IV}$ $P_j = \mathrm{D}(K, C_j) \oplus C_{j-1}\, j = 2, \ldots, N$

Advantages	Disadvantages
More secure than ECB for encrypting long plaintext — identical plaintext blocks produce different ciphertext.	Cannot be parallelized for encryption — each block depends on the previous ciphertext block and also cause slower in high-performance applications.
	Any error in encryption in any of blocks of ciphertexts is going to propagate due to each block depends on the previous ciphertext block
	Like ECB, plaintext must be padded to fit block size — adds overhead and possible leakage if padding is predictable.
	Needs both encryption and decryption functions — adds software/hardware complexity.
	Requires secure transmission or storage of the IV — improper handling can compromise security.

Course: Data Security
Lecturer: Dr. Mahmoud Yehia
TA: Eng. Eslam Osama
Block Cipher Operation Modes

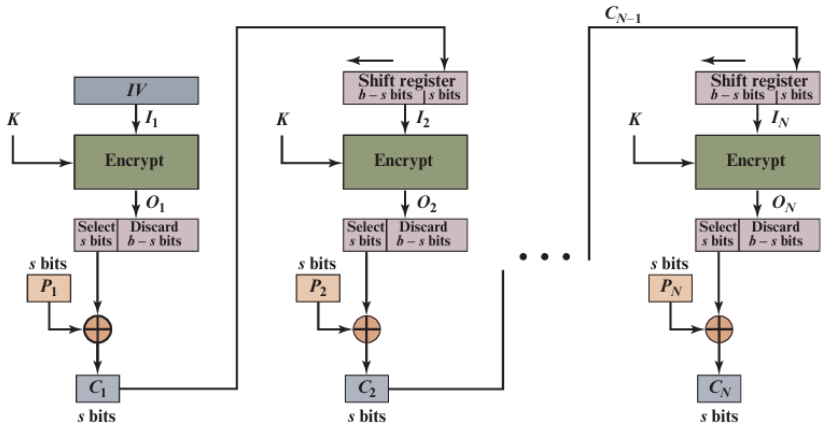
(3) Cipher Feedback Mode (CFB)

CFB mode allows a block cipher like AES or DES to work like a stream cipher, encrypting small chunks (like 8 bits) at a time instead of full blocks (like 128 or 64 bits). This is useful for real-time data (e.g., typing or streaming) and avoids padding the message.
In CFB:

- A shift register starts with an **initialization vector (IV)**.
- The cipher encrypts the register, then XORs part of the output with the plaintext to produce ciphertext.
- That ciphertext is then fed back into the register for the next round.
- Decryption uses the **same encryption function**, XORing again to get the original plaintext.

Advantages
Can operate in smaller units (bytes), making it suitable for streaming data
uses the **encryption function** only, simplifying hardware/software design.
The use of an Initialization Vector (IV) adds randomness to the encryption process, enhancing security.
Can be used in real-time applications as it is suitable for real time encryption

Disadvantages
Encryption is not parallelizable: Each segment/block of plaintext depends on the output of the previous encryption step.
Error propagation: A single bit error in the ciphertext affects the current and **next few** decrypted segments
Requires secure transmission or storage of the IV — improper handling can compromise security.



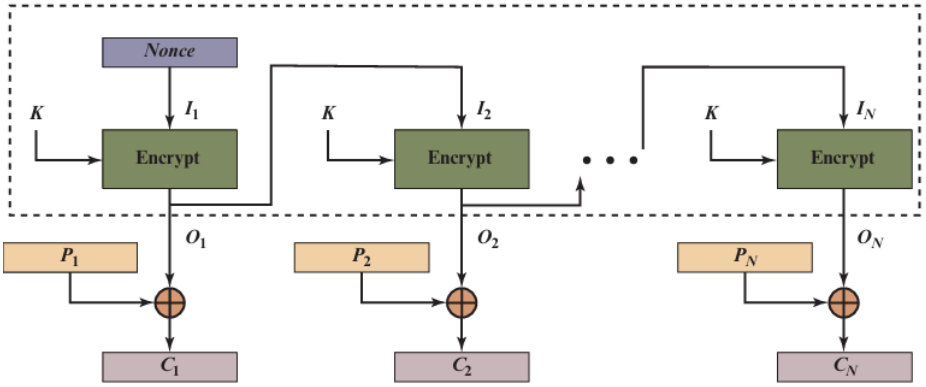
CFB	$I_1 = IV$ $I_j = \text{LSB}_{b-s}(I_{j-1}) \| C_{j-1}$ $j = 2, \ldots, N$ $O_j = \text{E}(K, I_j)$ $j = 1, \ldots, N$ $C_j = P_j \oplus \text{MSB}_s(O_j)$ $j = 1, \ldots, N$	$I_1 = IV$ $I_j = \text{LSB}_{b-s}(I_{j-1}) \| C_{j-1}$ $j = 2, \ldots, N$ $O_j = \text{E}(K, I_j)$ $j = 1, \ldots, N$ $P_j = C_j \oplus \text{MSB}_s(O_j)$ $j = 1, \ldots, N$

(4) Output Feedback Mode (OFB)

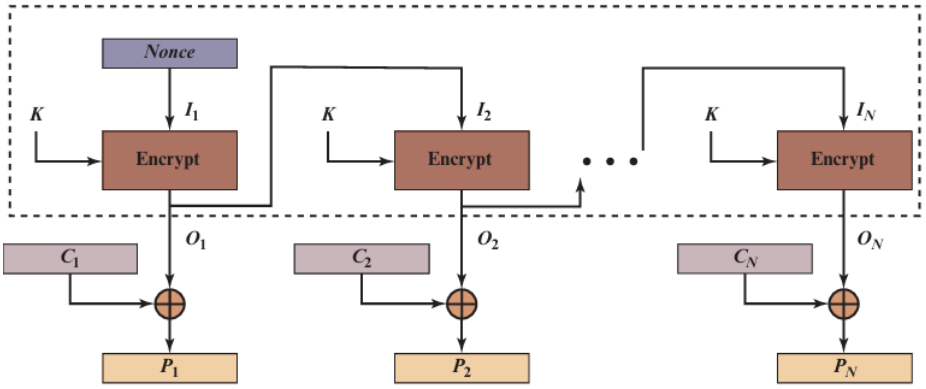
It is similar to Cipher Feedback (CFB) mode, but the main difference lies in what gets fed back into the encryption process. In CFB, the next input to the encryption function comes from the previous ciphertext, while in OFB, it comes from the previous encrypted output itself. This means OFB uses the output of the encryption function (not the ciphertext) to generate the next input, making it more resistant to transmission errors. Like CFB, OFB also uses only the encryption function for both encryption and decryption.

Advantages
Can operate in smaller units (bytes), making it suitable for streaming data
uses the **encryption function** only, simplifying hardware/software design.
Errors do not propagate: A bit error in the ciphertext affects **only the corresponding bit** in the plaintext during decryption.
Preprocessing is possible: The keystream can be generated in advance, even before the plaintext is available — useful in constrained or real-time systems.

Disadvantages
Not parallelizable: Like CFB, the keystream generation is sequential
Keystream reuse is catastrophic



(a) Encryption



(b) Decryption

OFB	$I_1 = Nonce$ $I_j = O_{j-1}$ $j = 2, \ldots, N$ $O_j = \mathrm{E}(K, I_j)$ $j = 1, \ldots, N$ $C_j = P_j \oplus O_j$ $j = 1, \ldots, N-1$ $C_N^* = P_N^* \oplus \mathrm{MSB}_u(O_N)$	$I_1 = Nonce$ $I_j = O_{j-1}$ $j = 2, \ldots, N$ $O_j = \mathrm{E}(K, I_j)$ $j = 1, \ldots, N$ $P_j = C_j \oplus O_j$ $j = 1, \ldots, N-1$ $P_N^* = C_N^* \oplus \mathrm{MSB}_u(O_N)$

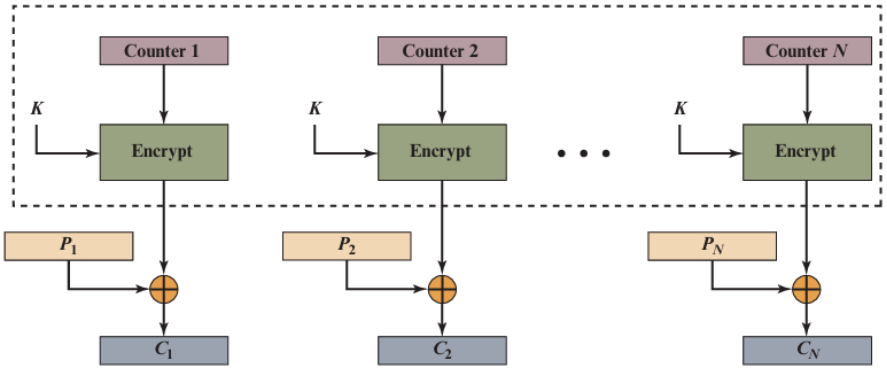
Note that, Nonce (Number used once) is used to prevent **replay attacks** and ensure **uniqueness** in modes like CTR

(4) Counter Mode (CTR)

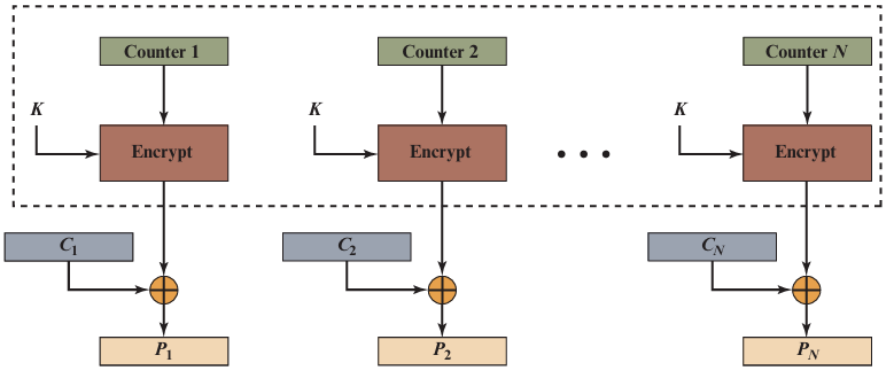
It is an encryption mode that turns a block cipher like AES into a stream cipher. Instead of encrypting the plaintext directly, CTR mode encrypts a counter value and then XORs the result with the plaintext to produce the ciphertext. It can handle any size of data (even 8-bit chunks), doesn't require padding, and allows fast parallel encryption and decryption using only the encryption function.

Advantages
Can operate in smaller units (bytes), making it suitable for streaming data
uses the **encryption function** only, simplifying hardware/software design.
Errors do not propagate: A bit error in the ciphertext affects **only the corresponding bit** in the plaintext during decryption.
Preprocessing is possible: The keystream can be generated in advance, even before the plaintext is available — useful in constrained or real-time systems.
Can be parallelized due to the use of different input to encrypt function in Encryption and decryption

Disadvantages
Keystream reuse is catastrophic



(a) Encryption



(b) Decryption

CTR	$C_j = P_j \oplus \mathrm{E}(K, T_j) \quad j = 1, \ldots, N-1$ $C_N^* = P_N^* \oplus \mathrm{MSB}_u[\mathrm{E}(K, T_N)]$	$P_j = C_j \oplus \mathrm{E}(K, T_j) \quad j = 1, \ldots, N-1$ $P_N^* = C_N^* \oplus \mathrm{MSB}_u[\mathrm{E}(K, T_N)]$