

Block Cipher Modes of Operation

1. Electronic Codebook (ECB)
2. Cipher Block Chaining (CBC)
3. Cipher Feedback (CFB)
4. Output Feedback (OFB)
5. Counter (CTR)

LIKE



COMMENT



SHARE



SUBSCRIBE



1. Electronic Codebook

⊗ 1 block @ a time

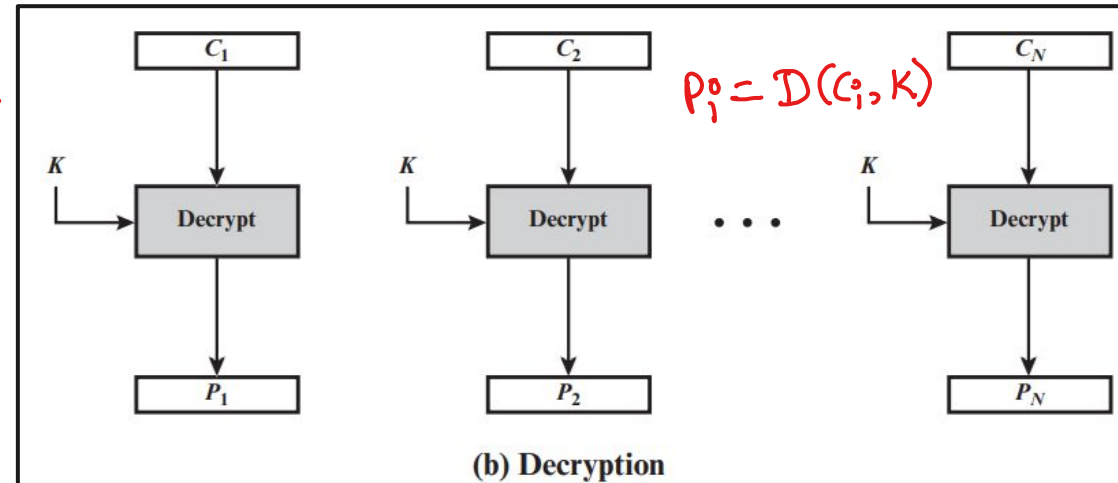
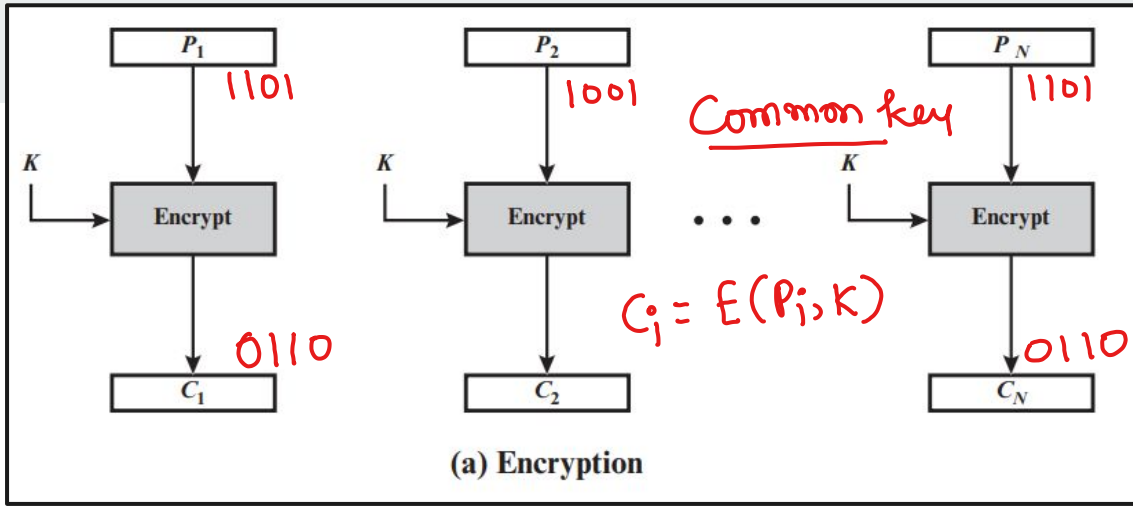
10char

$B \Rightarrow$ 8char


1001010010



padding



Key Points

- 
- Plaintext is handled one block at a time
 - Each block of plaintext is encrypted using the same key.
 - The term codebook is used because, for a given key, there is a unique ciphertext for every b -bit block of plaintext.
 - For a message longer than b bits, the procedure is simply to break the message into b -bit blocks, padding the last block if necessary.
 - Decryption is performed one block at a time, always using the same key.
 - If the same b -bit block of plaintext appears more than once in the message, it always produces the same ciphertext.

Criteria for other modes

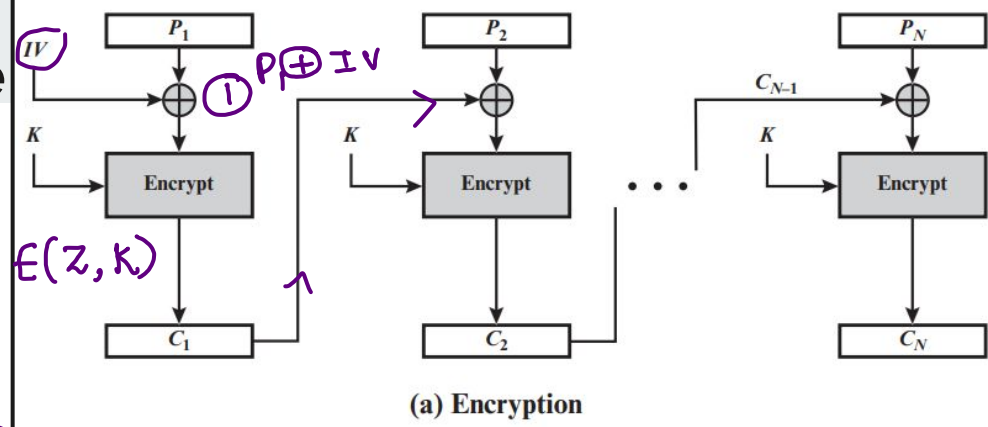


DOES

- **Overhead**: The additional operations for the encryption and decryption operation when compared to encrypting and decrypting in the ECB mode.
- **Error recovery**: The property that an error in the i th ciphertext block is inherited by only a few plaintext blocks after which the mode resynchronizes.
- **Error propagation**: The property that an error in the i th ciphertext block is inherited by the i th and all subsequent plaintext blocks. What is meant here is a bit error that occurs in the transmission of a ciphertext block, not a computational error in the encryption of a plaintext block.
- **Diffusion**: How the plaintext statistics are reflected in the ciphertext. Low entropy plaintext blocks should not be reflected in the ciphertext blocks. Roughly, low entropy equates to predictability or lack of randomness (see Appendix F).
- **Security**: Whether or not the ciphertext blocks leak information about the plaintext blocks.

2. Cipher Block Chaining Mode

⇒ Initialization vector
(b/w sender & receiver)



E :-

1) ex-or IV

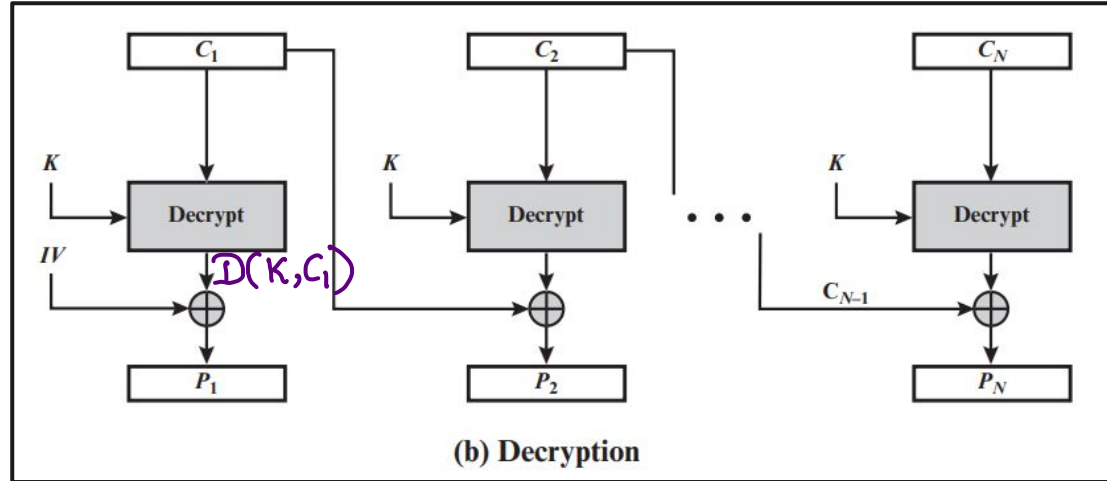
2) encrypt

D :- 1) decrypt

2) IV ex-or

$$C_i = E(K, P_i \oplus IV)$$

$$P_i = IV \oplus D(K, C_i)$$



Key Points

1. In this scheme, the input to the encryption algorithm is the XOR of the current plaintext block and the preceding ciphertext block; the same key is used for each block.
2. In effect, we have chained together the processing of the sequence of plaintext blocks.
3. It requires that the last block be padded to a full b bits if it is a partial block.
4. For decryption, each cipher block is passed through the decryption algorithm. The result is XORed with the preceding ciphertext block to produce the plaintext block
5. To produce the first block of ciphertext, an initialization vector (IV) is XORed with the first block of plaintext. On decryption, the IV is XORed with the output of the decryption algorithm to recover the first block of plaintext. The IV is a data block that is the same size as the cipher block.
6. IV confidentiality shared between sender and receiver.
7. In particular, for any given plaintext, it must not be possible to predict the IV that will be associated to the plaintext in advance of the generation of the IV.
8. because of the chaining mechanism of CBC, it is an appropriate mode for encrypting messages of length greater than b bits.

THANK YOU!

LIKE



COMMENT



SHARE



SUBSCRIBE



Convert Block to Stream Cipher

Why?

hi how are

- Eliminates need to pad data. ✓
- Encryption and decryption can be done on the fly. (no wait)

1. Cipher Feedback (CFB)
2. Output Feedback (OFB)
3. Counter (CTR)

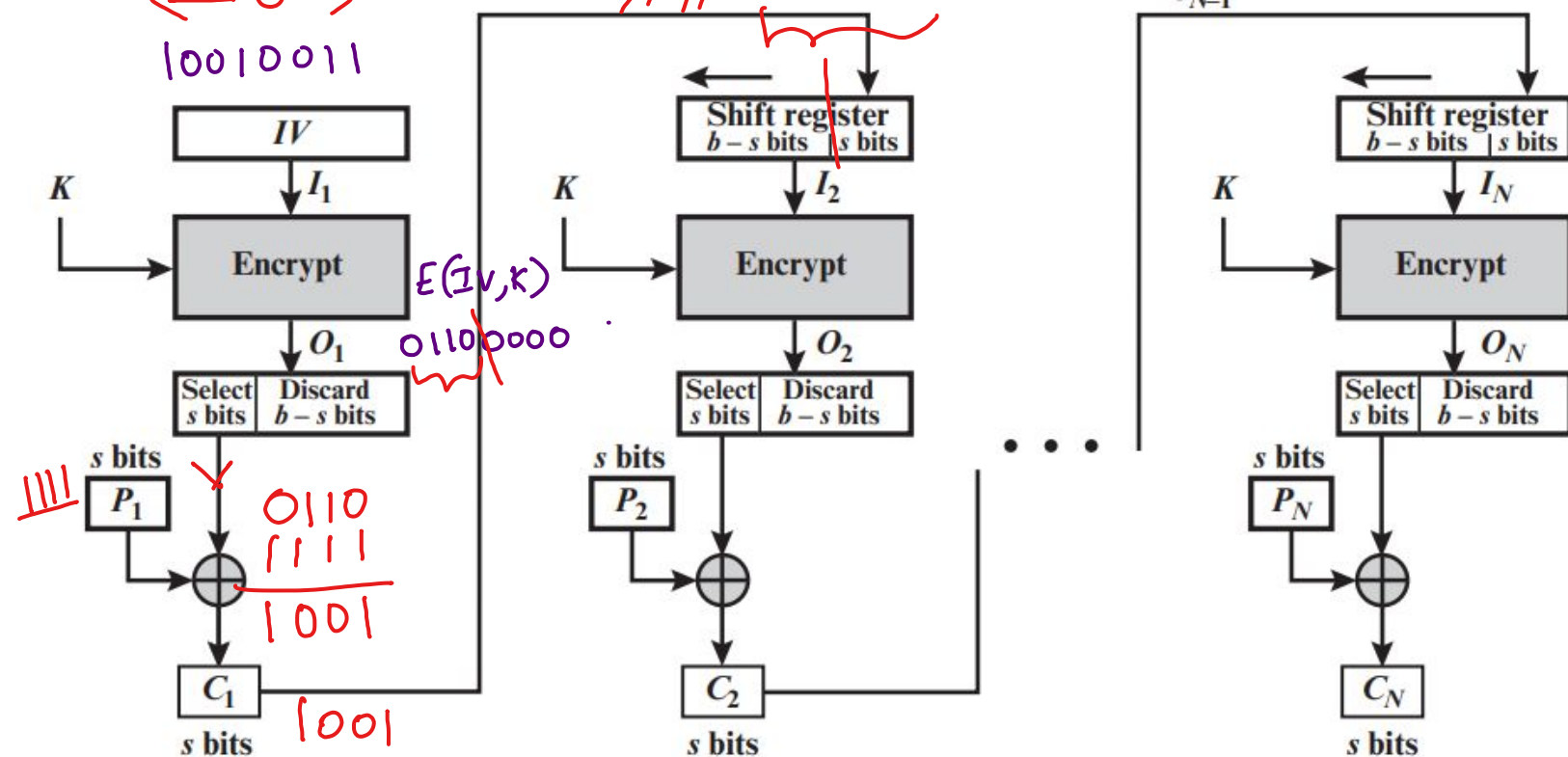
3. Cipher Feedback Mode

segments $\rightarrow s$ (4)

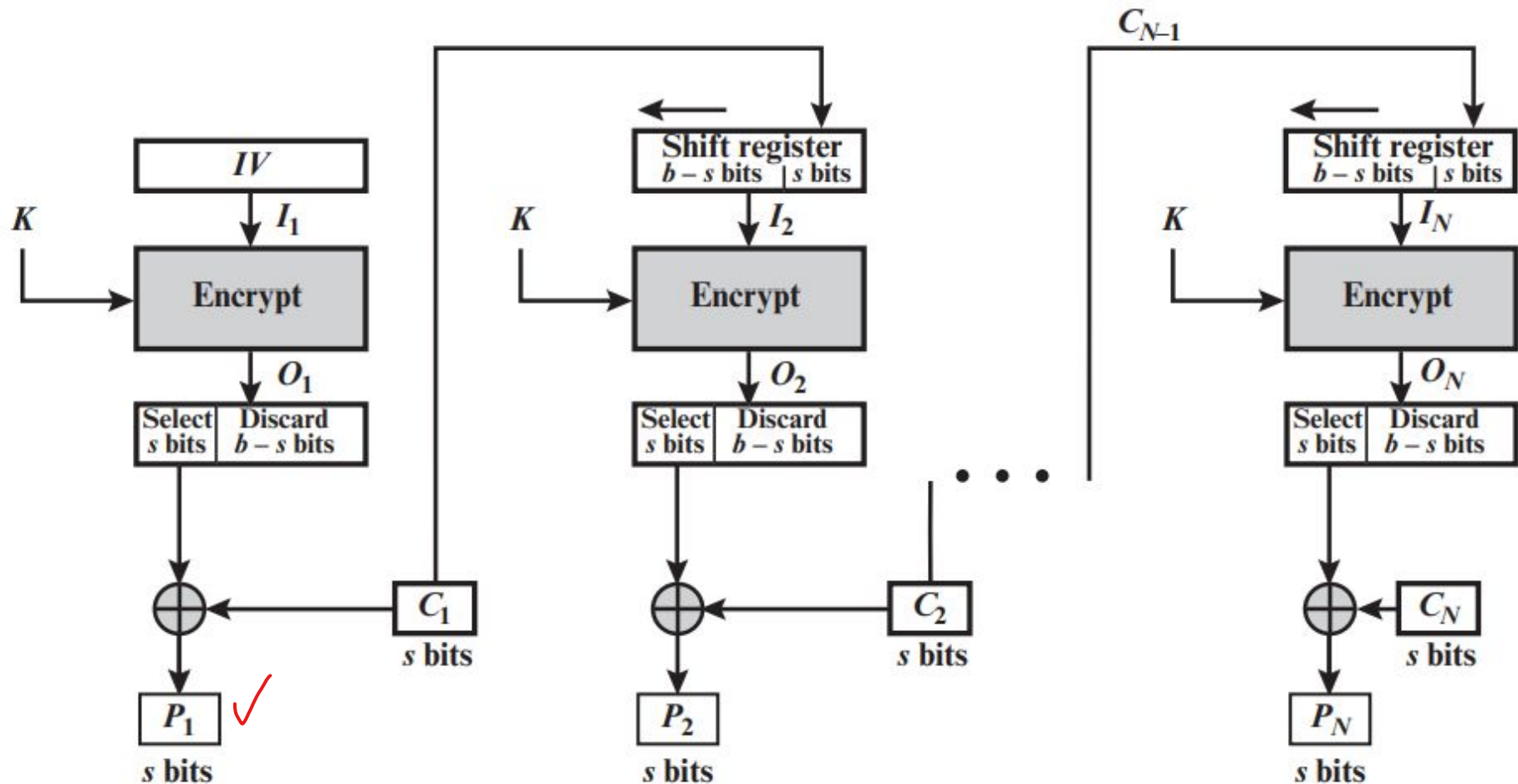
gen (8)

$s=4 \Rightarrow 8-4=4$
 $\leftarrow 6 \Rightarrow 8$
 10010011
 1111
 0110
 1111
 1001
 1001

Eg. $P \rightarrow (1111)0000$
 11110000
 11110000



(a) Encryption



(b) Decryption

Steps

The plaintext is divided into segments of s bits.

$$C_1 = P_1 \oplus \text{MSB}_s[E(K, IV)]$$

Encryption:

1. The input to the encryption function is a b -bit shift register that is initially set to some initialization vector (IV).
2. The leftmost (most significant) s bits of the output of the encryption function are XORed with the first segment of plaintext P_1 to produce the first unit of ciphertext C_1 , which is then transmitted.
3. In addition, the contents of the shift register are shifted left by s bits, and C_1 is placed in the rightmost (least significant) s bits of the shift register.
4. This process continues until all plaintext units have been encrypted.

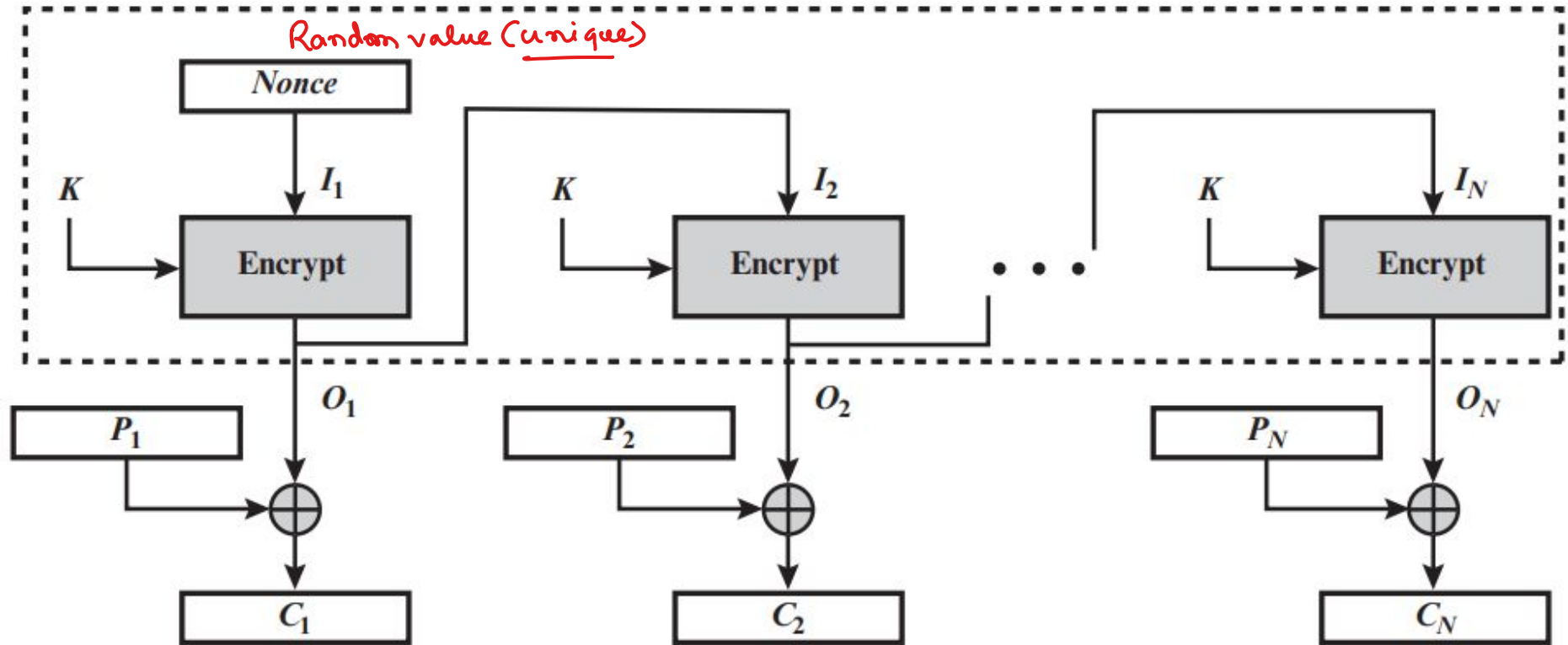
Decryption:

The same scheme is used, except that the received ciphertext unit is XORed with the output of the encryption function to produce the plaintext unit. Note that it is the encryption function that is used, not the decryption function.

$$P_1 = C_1 \oplus \text{MSB}_s[E(K, IV)]$$

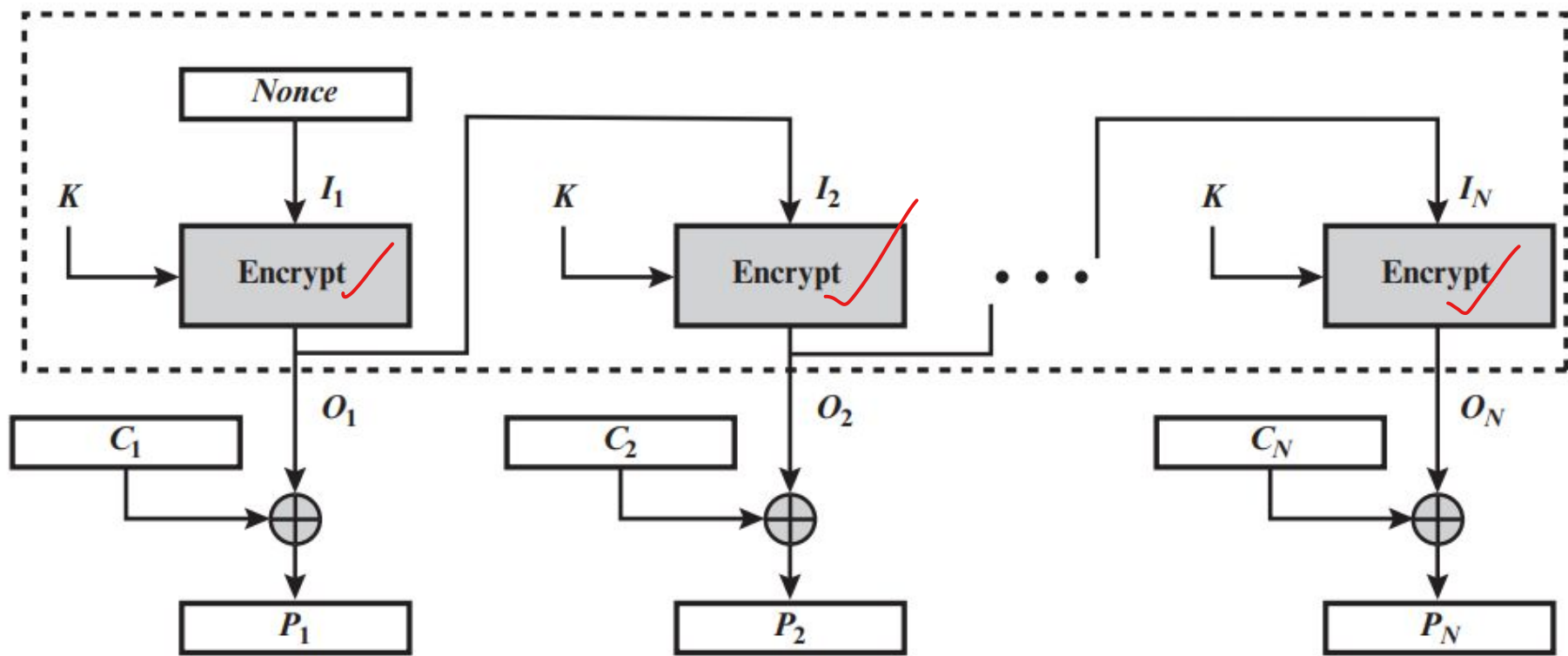
4. Output Feedback Mode

$$C_j = \underline{P_j} \oplus E(K, [C_{j-1} \oplus P_{j-1}])$$



(a) Encryption

$$P_j = C_j \oplus E(K, [C_{j-1} \oplus P_{j-1}])$$



(b) Decryption

Key Points

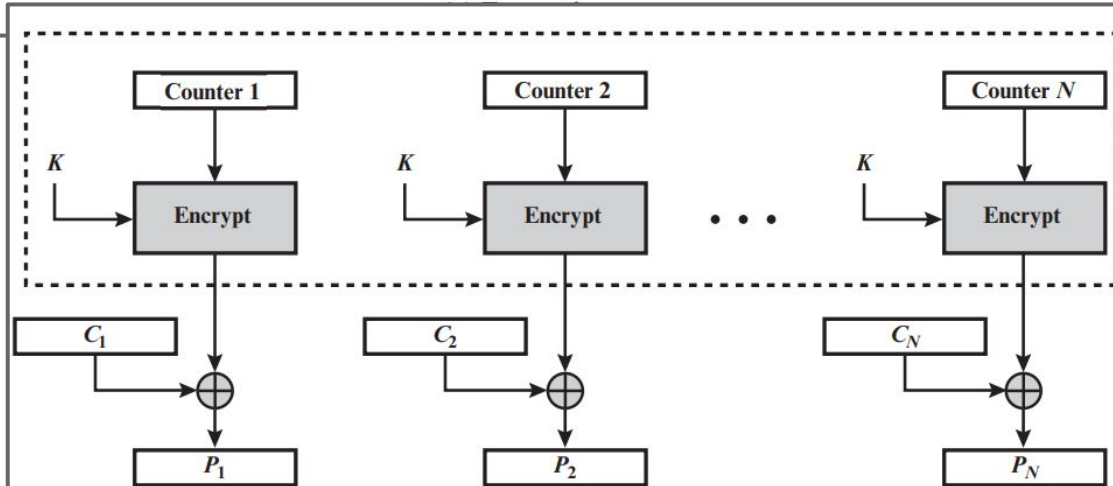
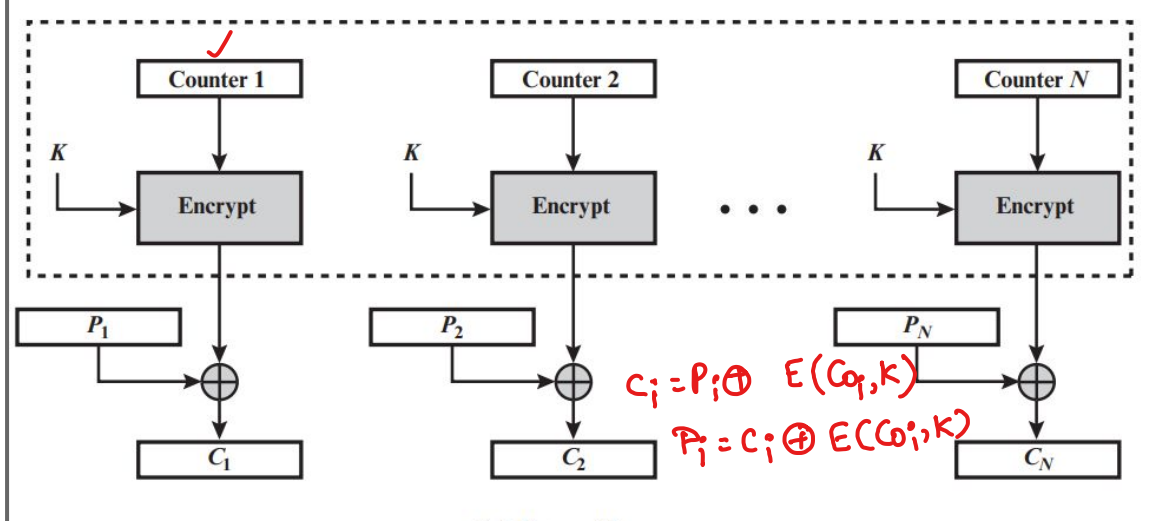
- For OFB, the output of the encryption function is fed back to become the input for encrypting the next block of plaintext
- OFB mode operates on full blocks of plaintext and ciphertext.
- IV(Nonce) must be unique to each execution of the encryption operation
- One advantage of the OFB method is that bit errors in transmission do not propagate.
- The disadvantage of OFB is that it is more vulnerable to a message stream modification attack than is CFB

4. Counter Mode

$P_1 \rightarrow 1234$

$P_2 \rightarrow 1235$

$P_3 \rightarrow \underline{1236}$



(b) Decryption

Key Points

A counter equal to the plaintext block size is used.

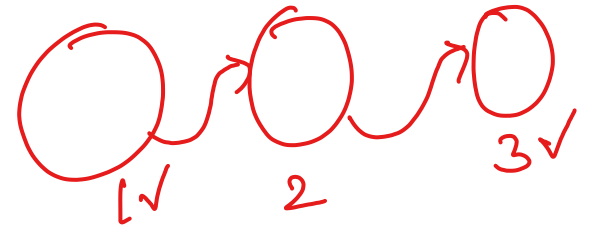
The counter value must be different for each plaintext block that is encrypted. Typically, the counter is initialized to some value and then incremented by 1 for each subsequent block .

For encryption, the counter is encrypted and then XORed with the plaintext block to produce the ciphertext block; there is no chaining.

For decryption, the same sequence of counter values is used, with each encrypted counter XORed with a ciphertext block to recover the corresponding plaintext block. Thus, the initial counter value must be made available for decryption.

For the last plaintext block, which may be a partial block of u bits, the most significant u bits of the last output block are used for the XOR operation; the remaining $b - u$ bits are discarded.

Advantages of Counter Mode



1. **Hardware efficiency:** Unlike the three chaining modes, encryption (or decryption) in CTR mode can be done in parallel on multiple blocks of plaintext or ciphertext.
2. **Software efficiency:** Similarly, because of the opportunities for parallel execution in CTR mode, processors that support parallel features, such as aggressive pipelining, multiple instruction dispatch per clock cycle, a large number of registers, and SIMD instructions, can be effectively utilized.
3. **Random access:** The i th block of plaintext or ciphertext can be processed in random-access fashion. With the chaining modes, block C_i cannot be computed until the $i - 1$ prior blocks are computed.
4. **Provable security:** It can be shown that CTR is at least as secure as the other modes discussed in this chapter.
5. **Simplicity:** Unlike ECB and CBC modes, CTR mode requires only the implementation of the encryption algorithm and not the decryption algorithm. This matters most when the decryption algorithm differs substantially from the encryption algorithm, as it does for AES.

THANK YOU!

LIKE



COMMENT



SHARE



SUBSCRIBE



Table 7.1 Block Cipher Modes of Operation

K P_1 K P_2

Mode	Description	Typical Application
Electronic Codebook (ECB)	Each block of plaintext bits is encoded independently using the same key.	<ul style="list-style-type: none"> Secure transmission of single values (e.g., an encryption key)
Cipher Block Chaining (CBC)	The input to the encryption algorithm is the XOR of the next block of plaintext and the preceding block of ciphertext. $P_2 \rightarrow C_1 P_2$	<ul style="list-style-type: none"> General-purpose block-oriented transmission Authentication
<u>Cipher Feedback (CFB)</u>	Input is processed s bits at a time. Preceding ciphertext is used as input to the encryption algorithm to produce pseudorandom output, which is XORed with plaintext to produce next unit of ciphertext.	<ul style="list-style-type: none"> General-purpose stream-oriented transmission ✓ Authentication ✓
<u>Output Feedback (OFB)</u>	Similar to CFB, except that the input to the encryption algorithm is the preceding encryption output, and full blocks are used.	<ul style="list-style-type: none"> Stream-oriented transmission over noisy channel (e.g., satellite communication) ✓
<u>Counter (CTR)</u> ✓	Each block of plaintext is XORed with an encrypted counter. The counter is incremented for each subsequent block.	<ul style="list-style-type: none"> General-purpose block-oriented transmission Useful for high-speed requirements ✓