

**Symmetric Cryptography**

**Block Cipher Operation Modes**

**Sang-Yoon Chang, Ph.D.**

## **Modern Cipher (vs. Classical Cipher)**

Digital computer communications  
based on bits

Product cipher

More sophisticated techniques

## **Module Objectives:**

### **Block Cipher Operation Modes**

Electronic Codebook (ECB)

Cipher Block Chaining (CBC)

Cipher Feedback (CFB)

Output Feedback (OFB)

Counter (CTR)



## Block Cipher Operation Modes

Handle plaintext that can be longer than  
a block in a secure manner

Alice and Bob agrees on a key and a  
cipher (an enc./dec. function)

## Variables

$b$ : the block length

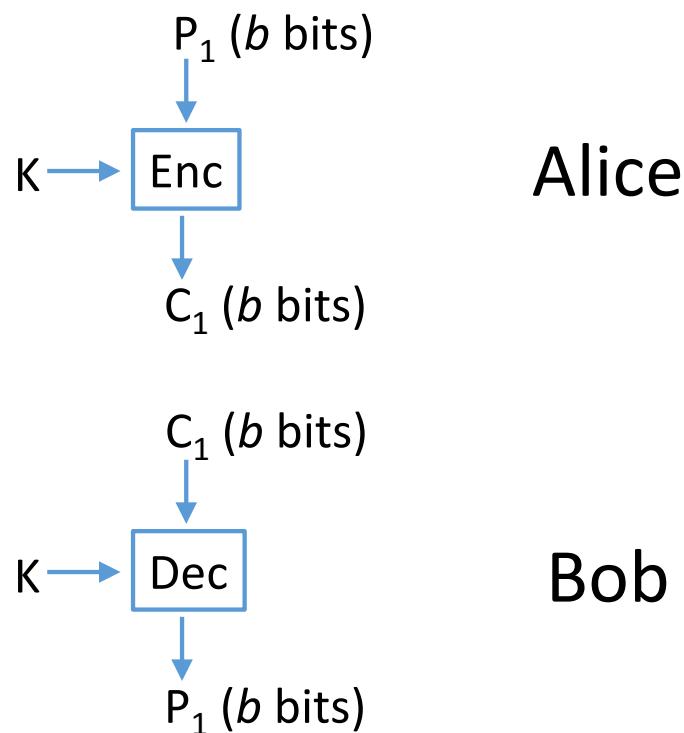
$P_i$ :  $i$ -th plaintext block ( $b$  bits)

$C_i$ :  $i$ -th ciphertext block ( $b$  bits)

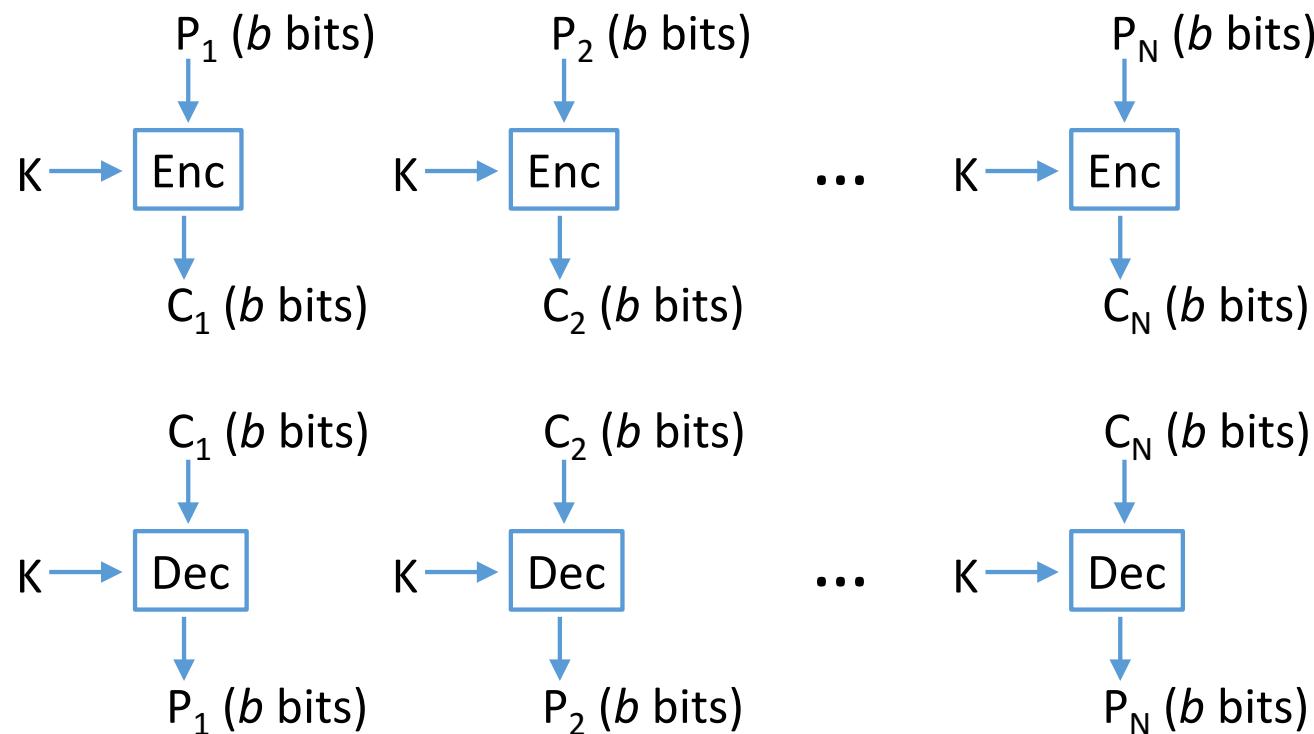
K: key

# **Electronic Code Book (ECB)**

## ECB Encryption (top) and Decryption (bottom)



## ECB Encryption (top) and Decryption (bottom)



## **Electronic Code Book (ECB)**

A simple block cipher mode

Use the same raw key over multiple  
blocks

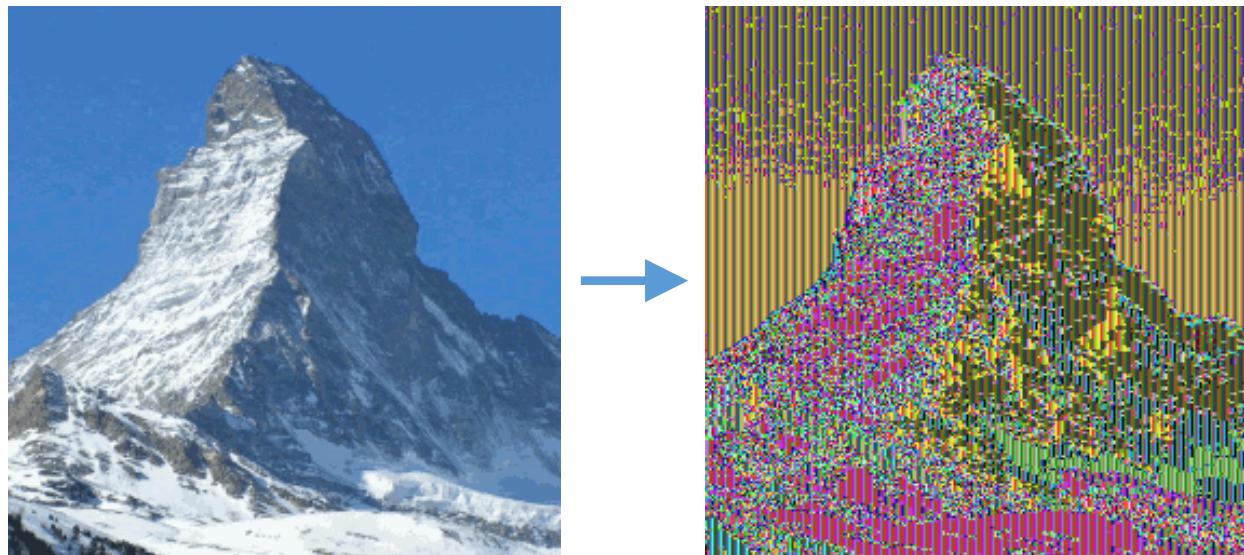
For block  $i$ ,  $C_i = \text{Enc}(K, P_i)$ ,  $P_i = \text{Dec}(K, C_i)$

## **Electronic Code Book Security**

Use the same raw key over multiple blocks

Redundancy/patterns in long plaintext  
carries over to the ciphertext

## Electronic Code Book Security



(Source: Dake from Wikimedia Commons)



## Cipher Block Chaining (CBC) Mode

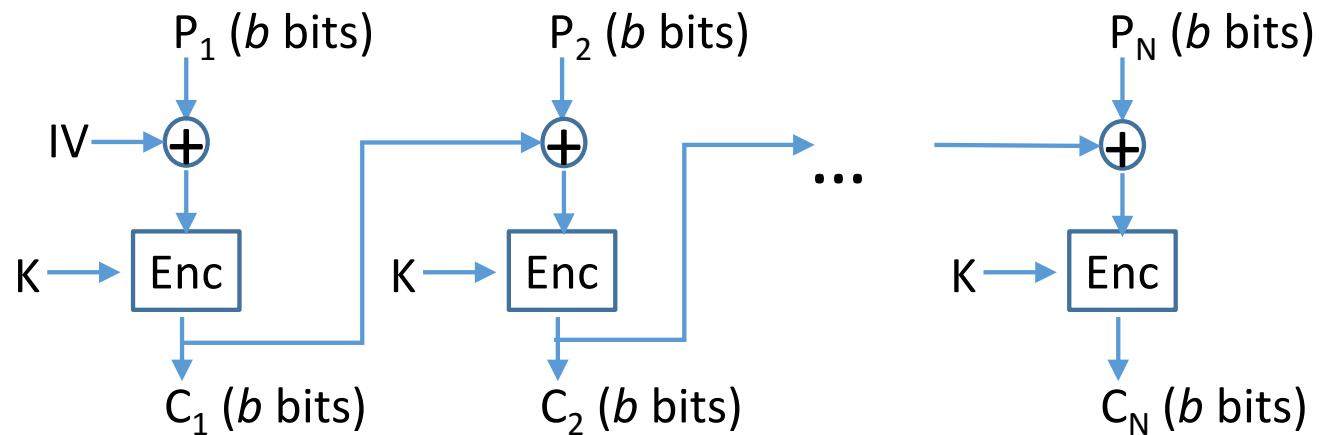
Each previous ciphertext block is chained with the current plaintext block

Applicable when all data is already available

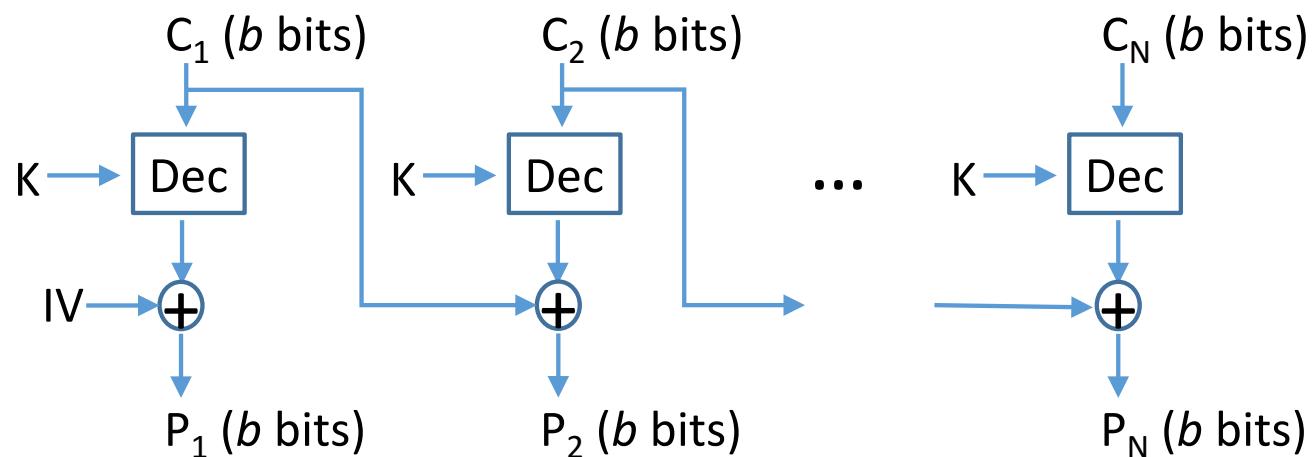
For block  $i$ ,

Encryption:  $C_i = \text{Enc}(K, [C_{i-1} \oplus P_i])$

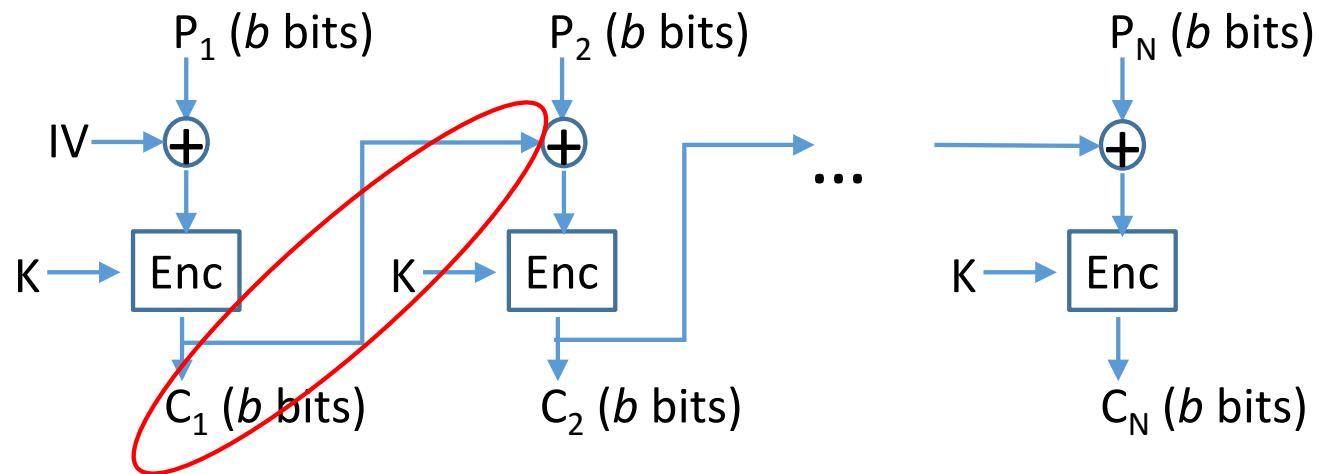
Decryption:  $P_i = \text{Dec}(K, C_i) \oplus C_{i-1}$



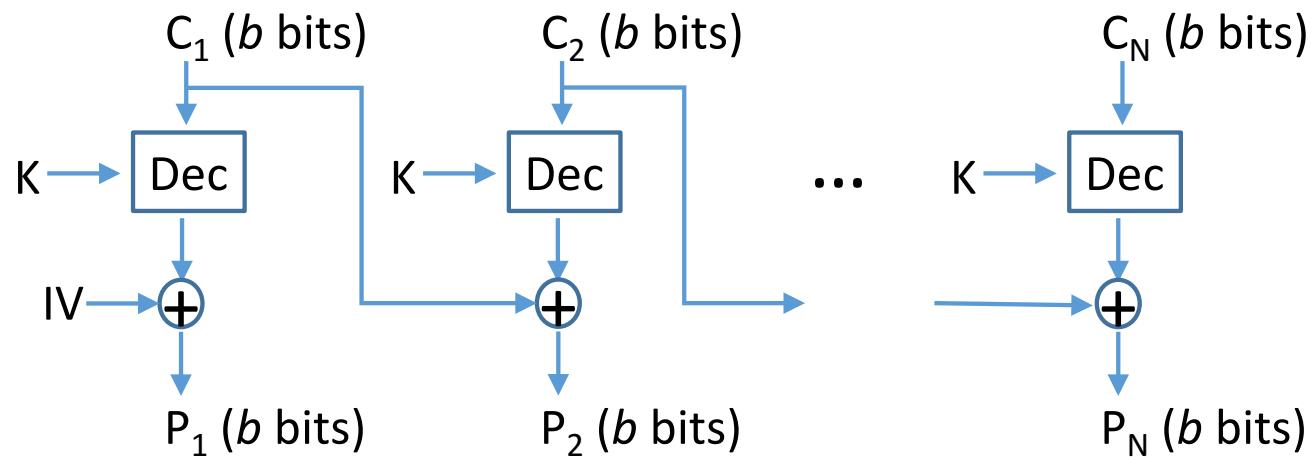
CBC Encryption



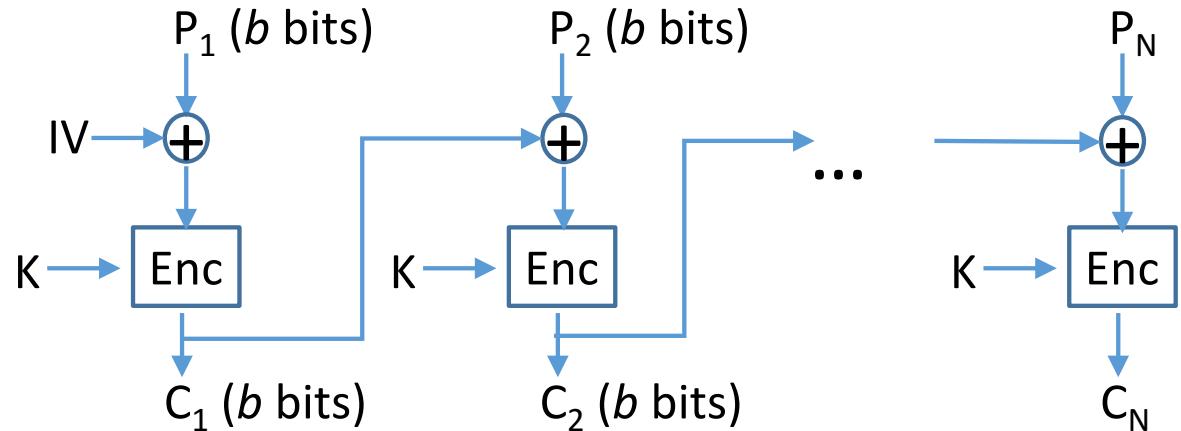
CBC Decryption



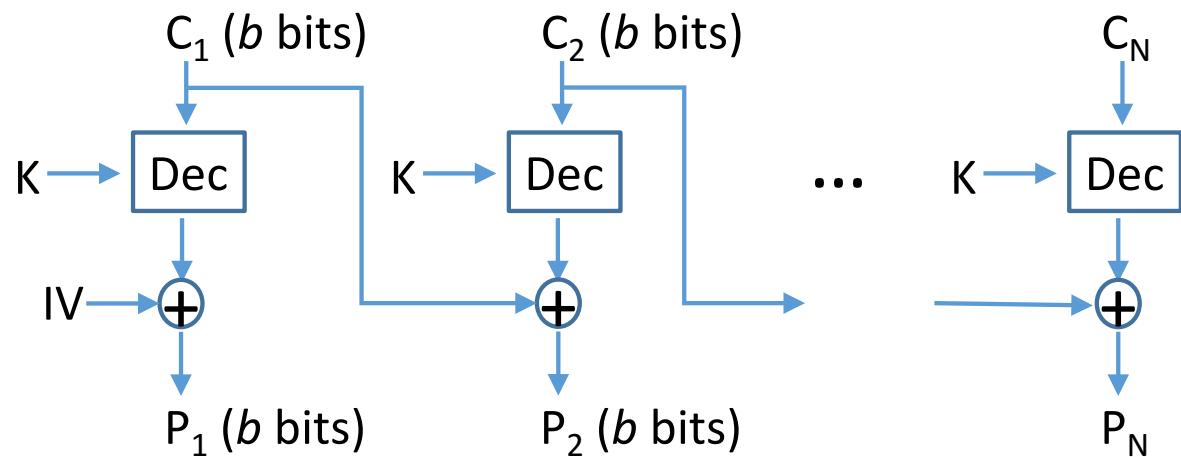
CBC Encryption



CBC Decryption

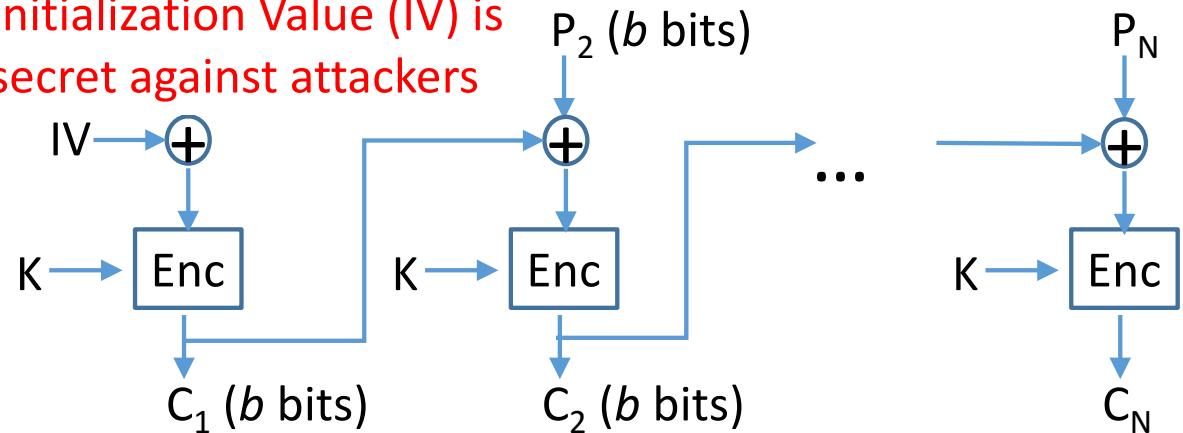


CBC Encryption  
 $C_i = \text{Enc}(K, [C_{i-1} \oplus P_i])$



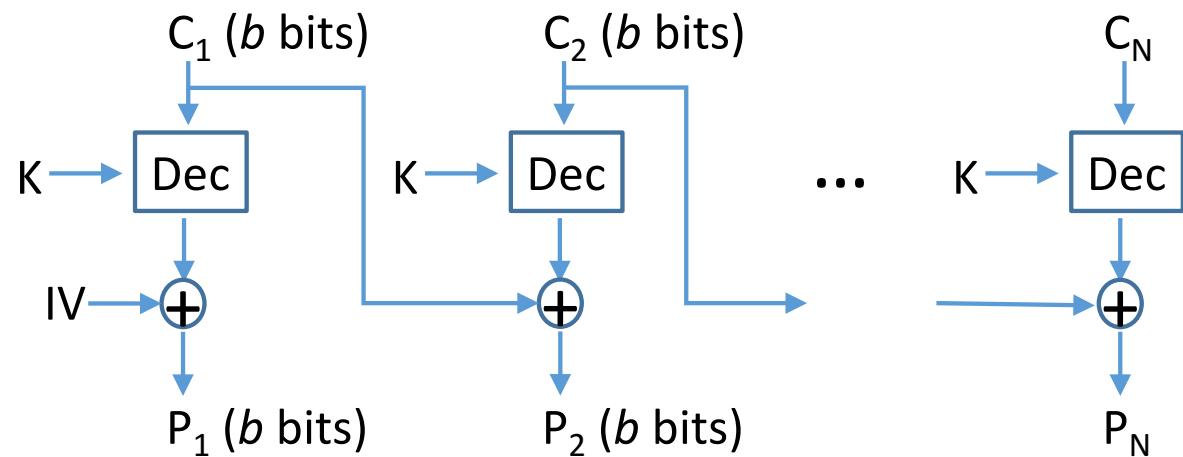
CBC Decryption  
 $P_i = \text{Dec}(K, C_i) \oplus C_{i-1}$

Initialization Value (IV) is secret against attackers



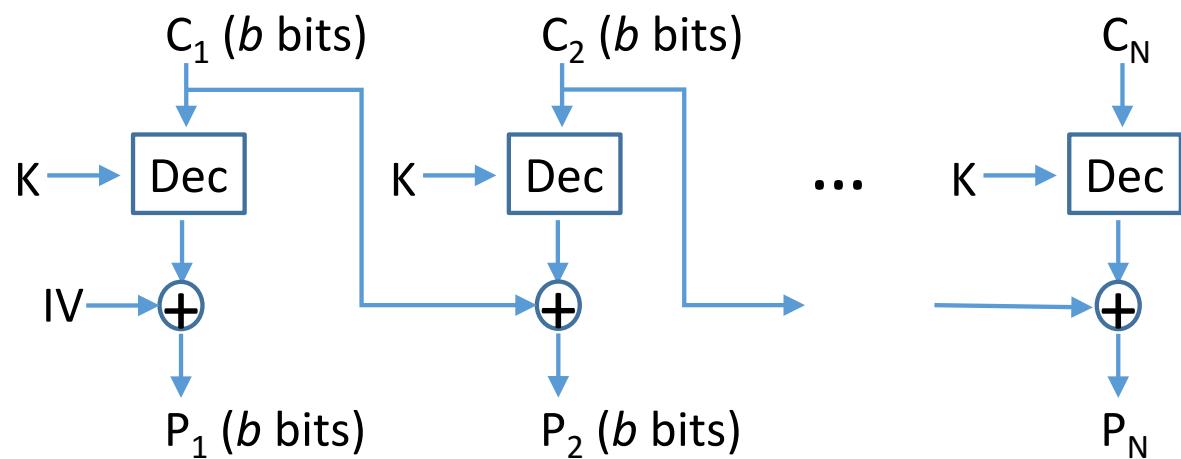
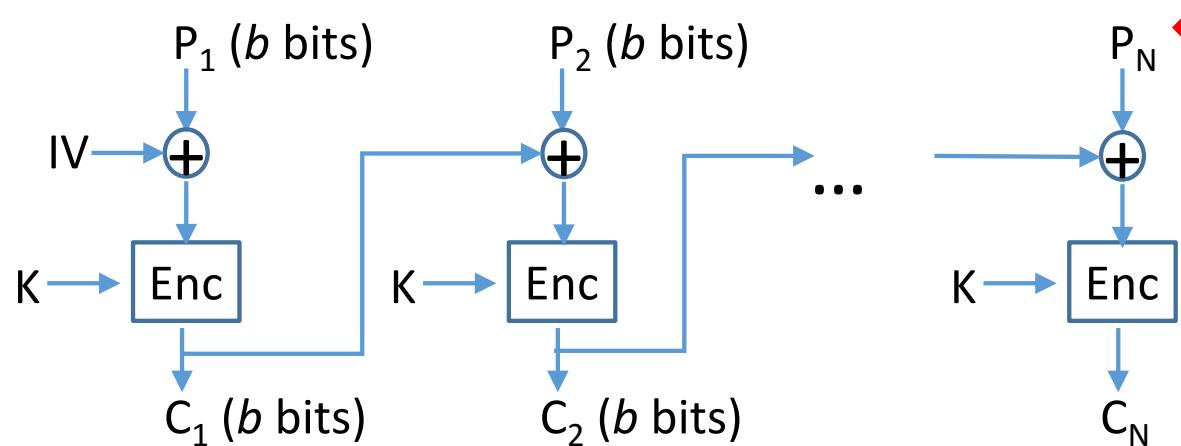
CBC Encryption

$$C_i = \text{Enc}(K, [C_{i-1} \oplus P_i])$$



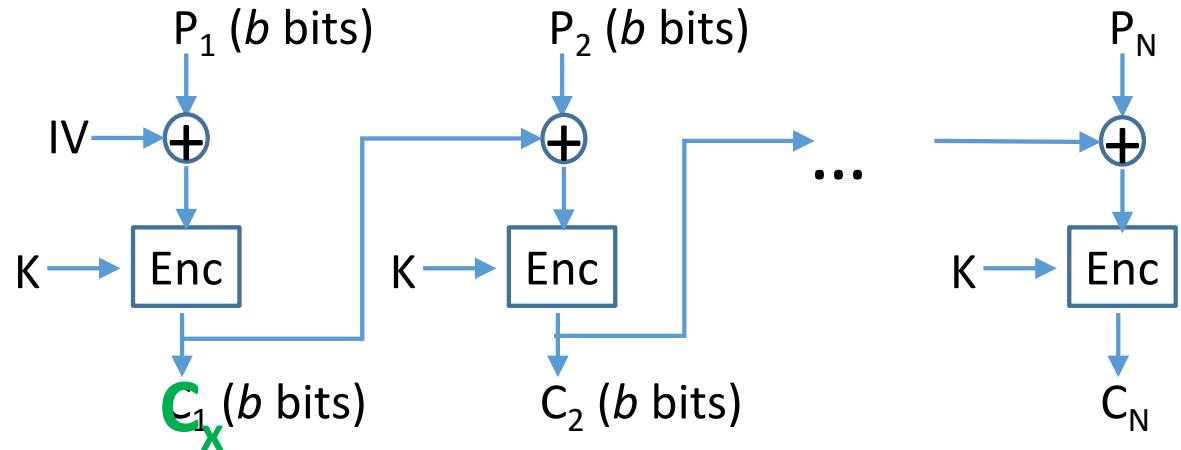
CBC Decryption

$$P_i = \text{Dec}(K, C_i) \oplus C_{i-1}$$

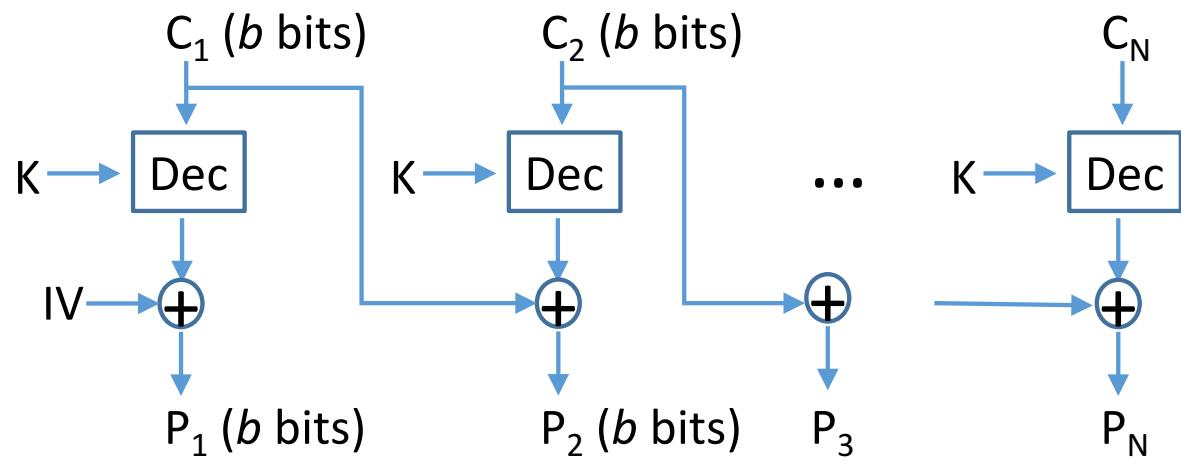


**Pad if incomplete block**  
**CBC Encryption**  
 $C_i = \text{Enc}(K, [C_{i-1} \oplus P_i])$

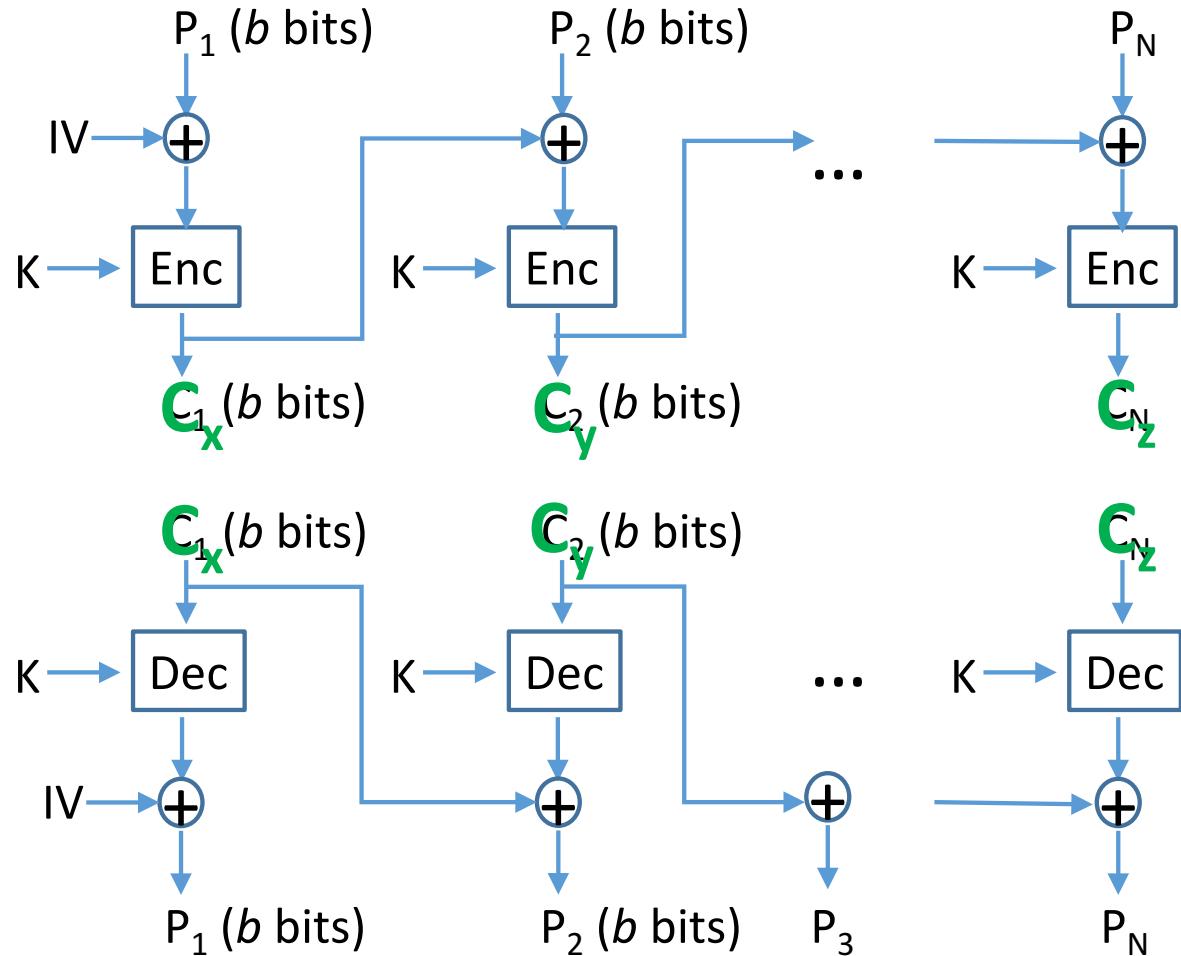
**CBC Decryption**  
 $P_i = \text{Dec}(K, C_i) \oplus C_{i-1}$



CBC Encryption  
 $C_i = \text{Enc}(K, [C_{i-1} \oplus P_i])$



CBC Decryption  
 $P_i = \text{Dec}(K, C_i) \oplus C_{i-1}$



CBC Encryption

$$C_i = \text{Enc}(K, [C_{i-1} \oplus P_i])$$

CBC Decryption

$$P_i = \text{Dec}(K, C_i) \oplus C_{i-1}$$



# **Stream Modes of Operation for Block Cipher**

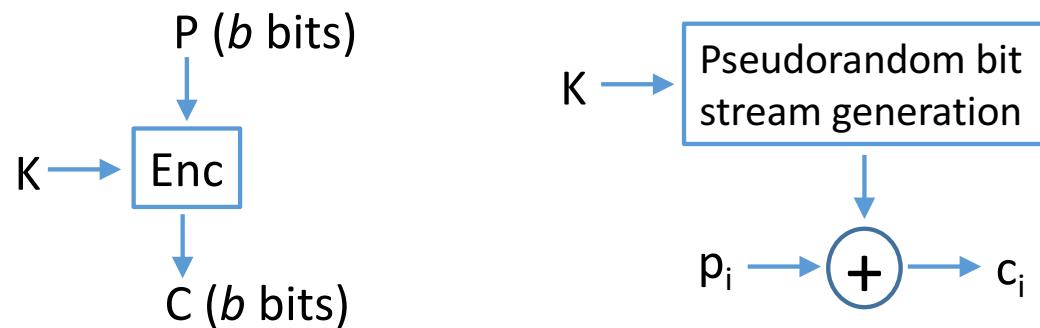
Cipher feedback (CFB) mode

Output feedback (OFB) mode

Counter (CTR) mode

## Block Cipher vs. Stream Cipher

Block cipher (left) processes in blocks (multiple bits) while stream cipher (right) processes them a bit/byte at a time



## **Stream Modes of Operation for Block Cipher**

Block cipher for pseudo-random generator

XOR the data (enables smaller unit)

CFB mode, OFB mode, and CTR mode

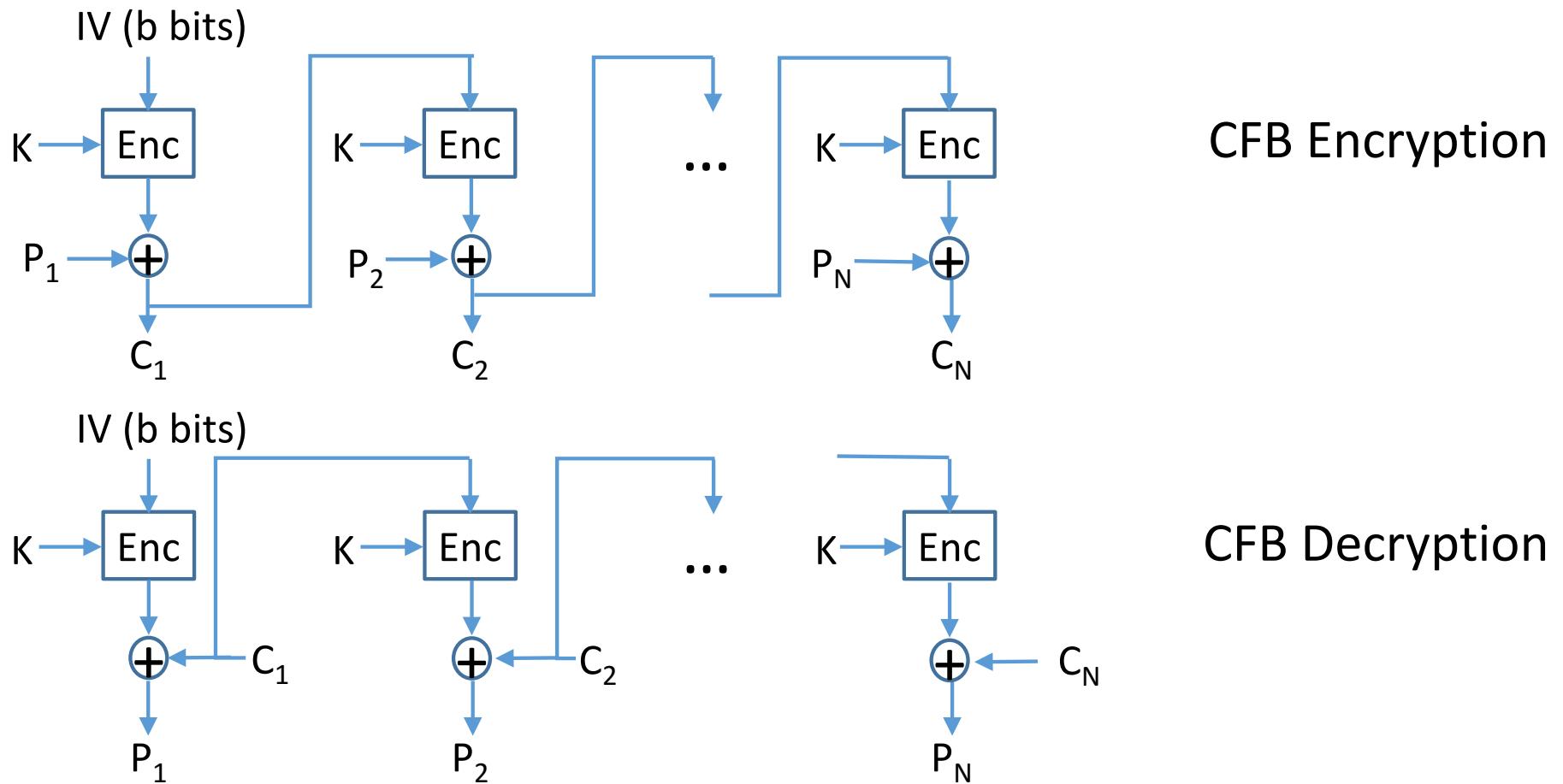
## Cipher Feedback (CFB) Mode

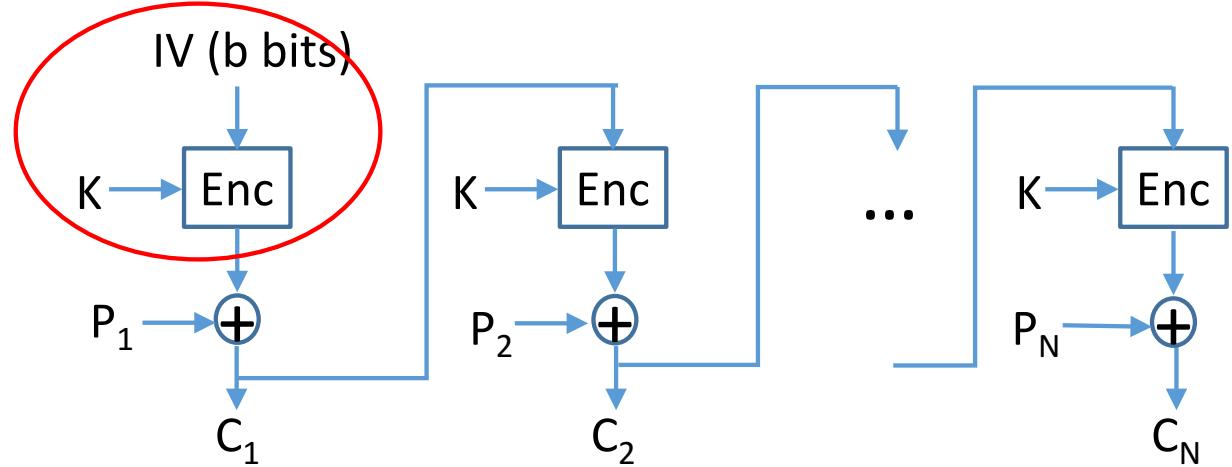
Use key to generate pseudo-random bits

Ciphertext is fed to the pseudo-random generator

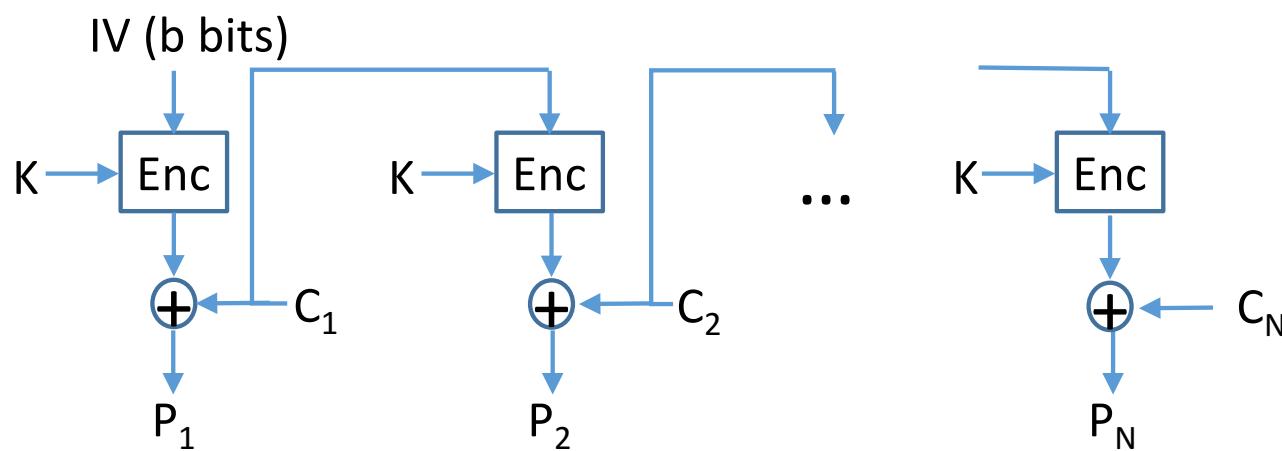
Encryption:  $C_i = P_i \oplus \text{Enc}(K, C_{i-1})$

Decryption:  $P_i = C_i \oplus \text{Enc}(K, C_{i-1})$

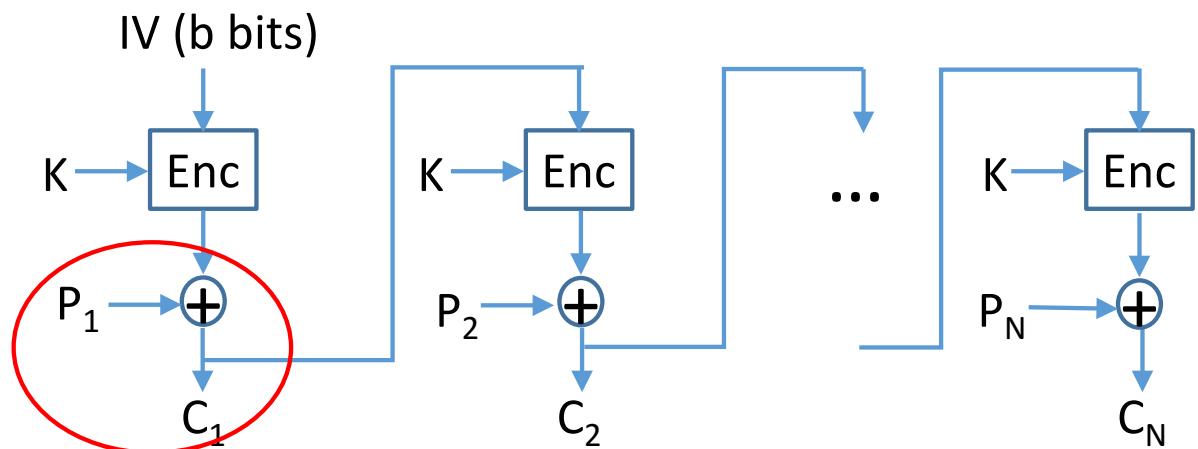




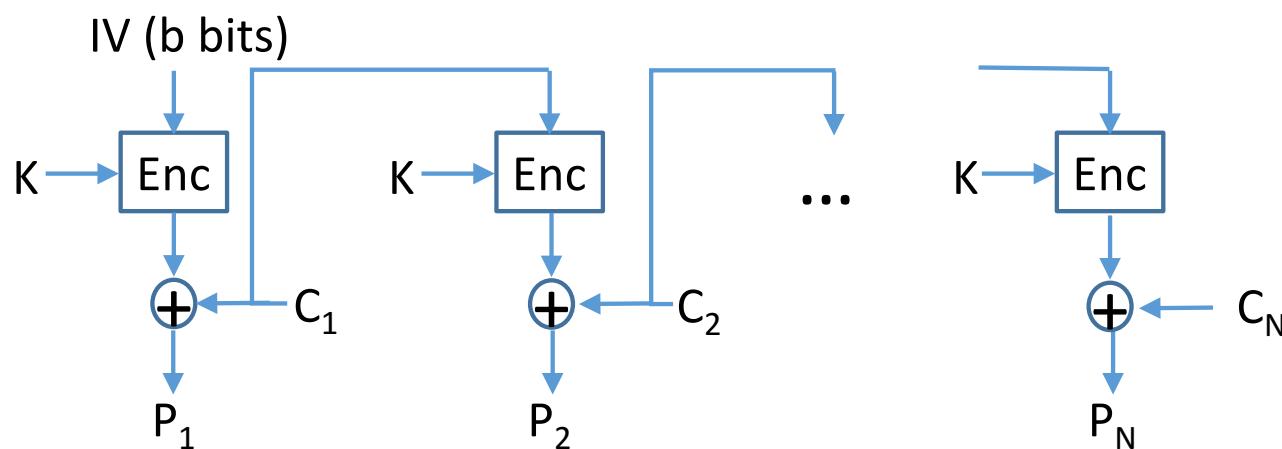
CFB Encryption



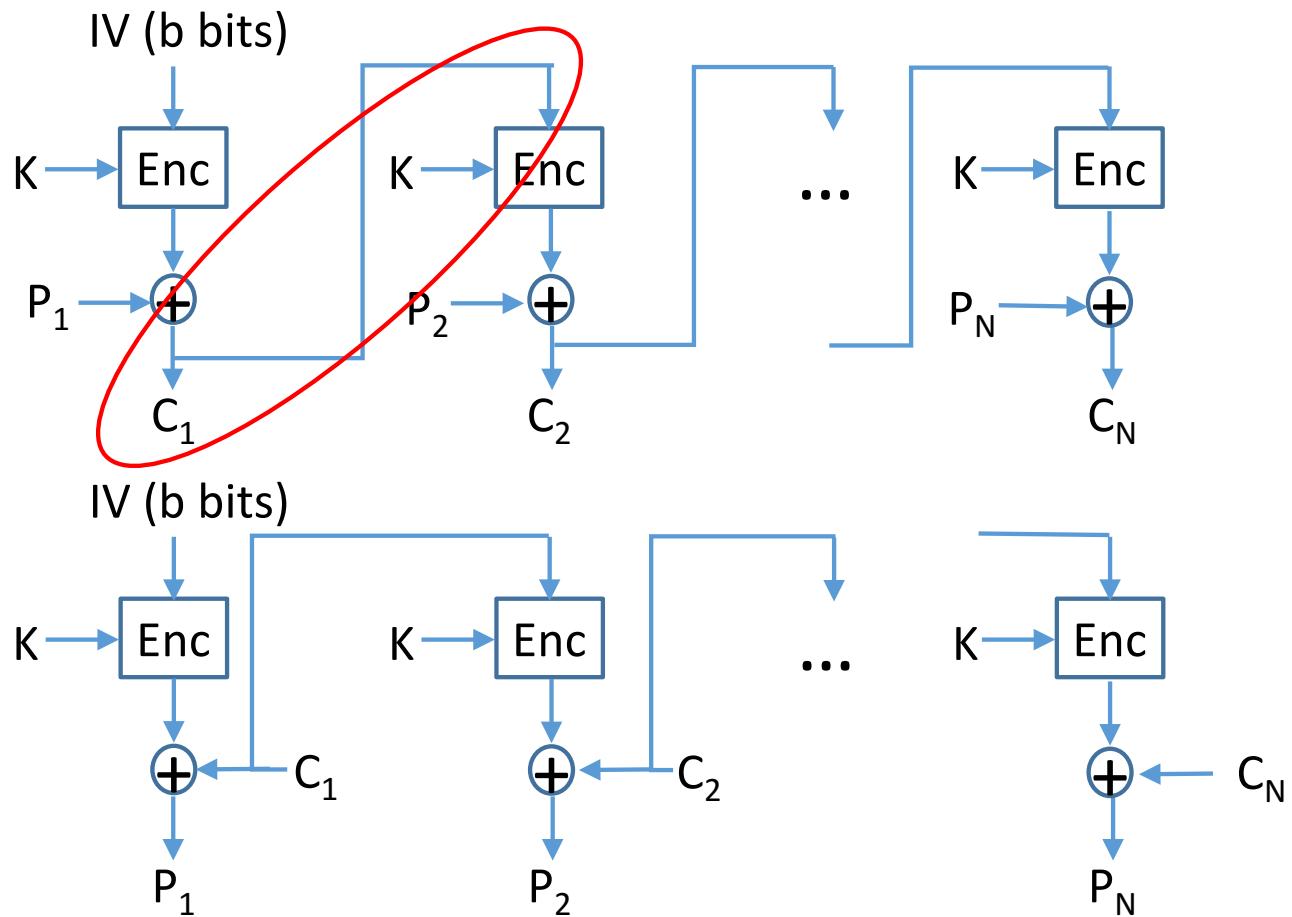
CFB Decryption



CFB Encryption

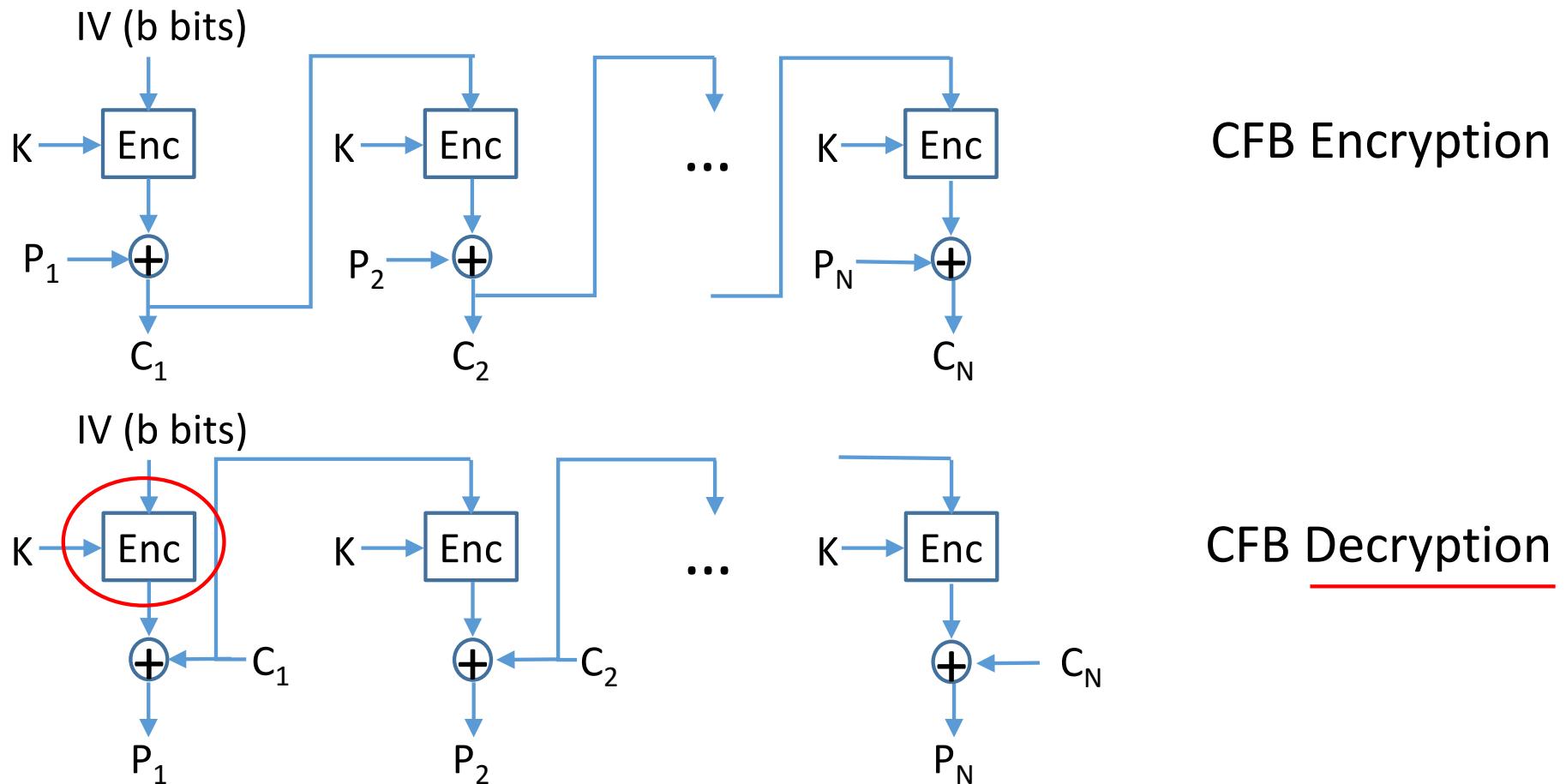


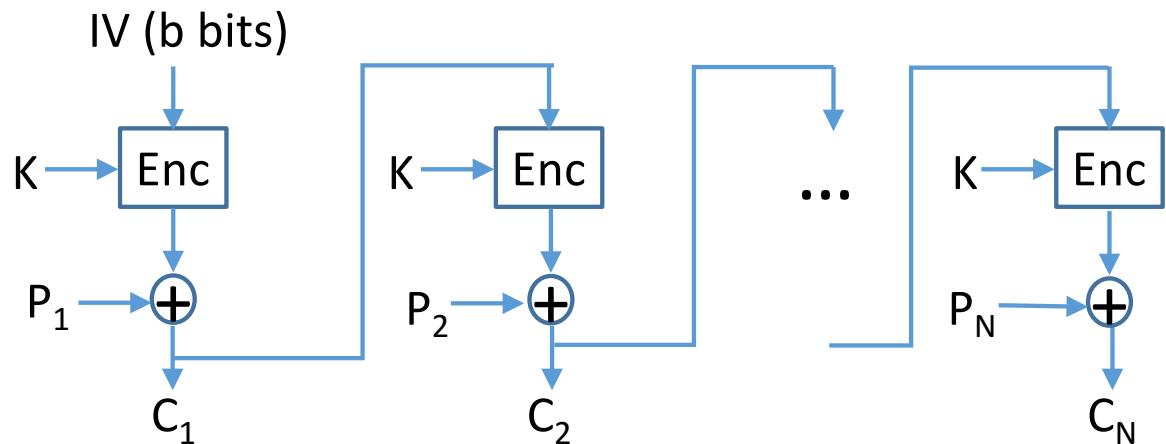
CFB Decryption



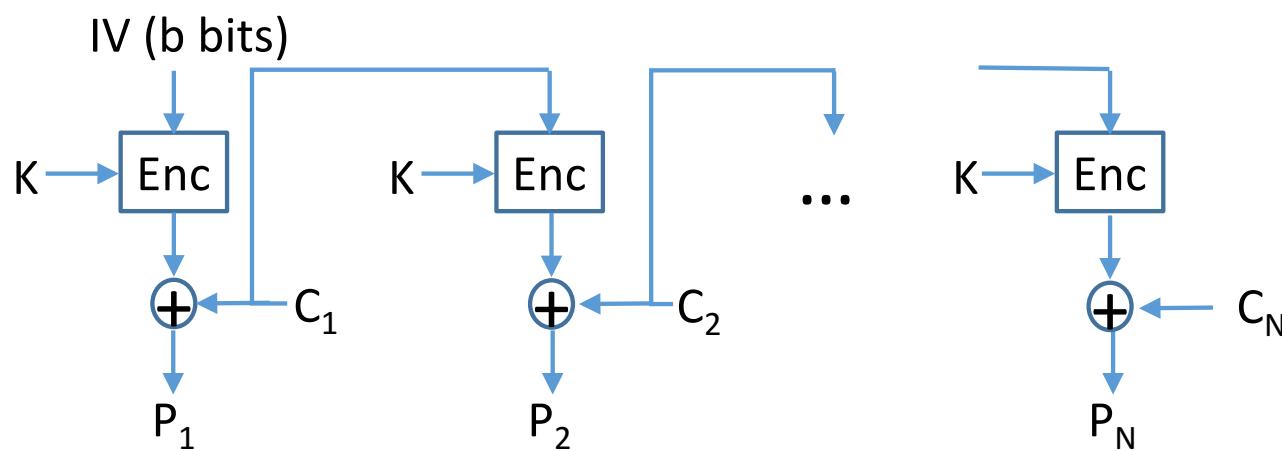
CFB Encryption

CFB Decryption

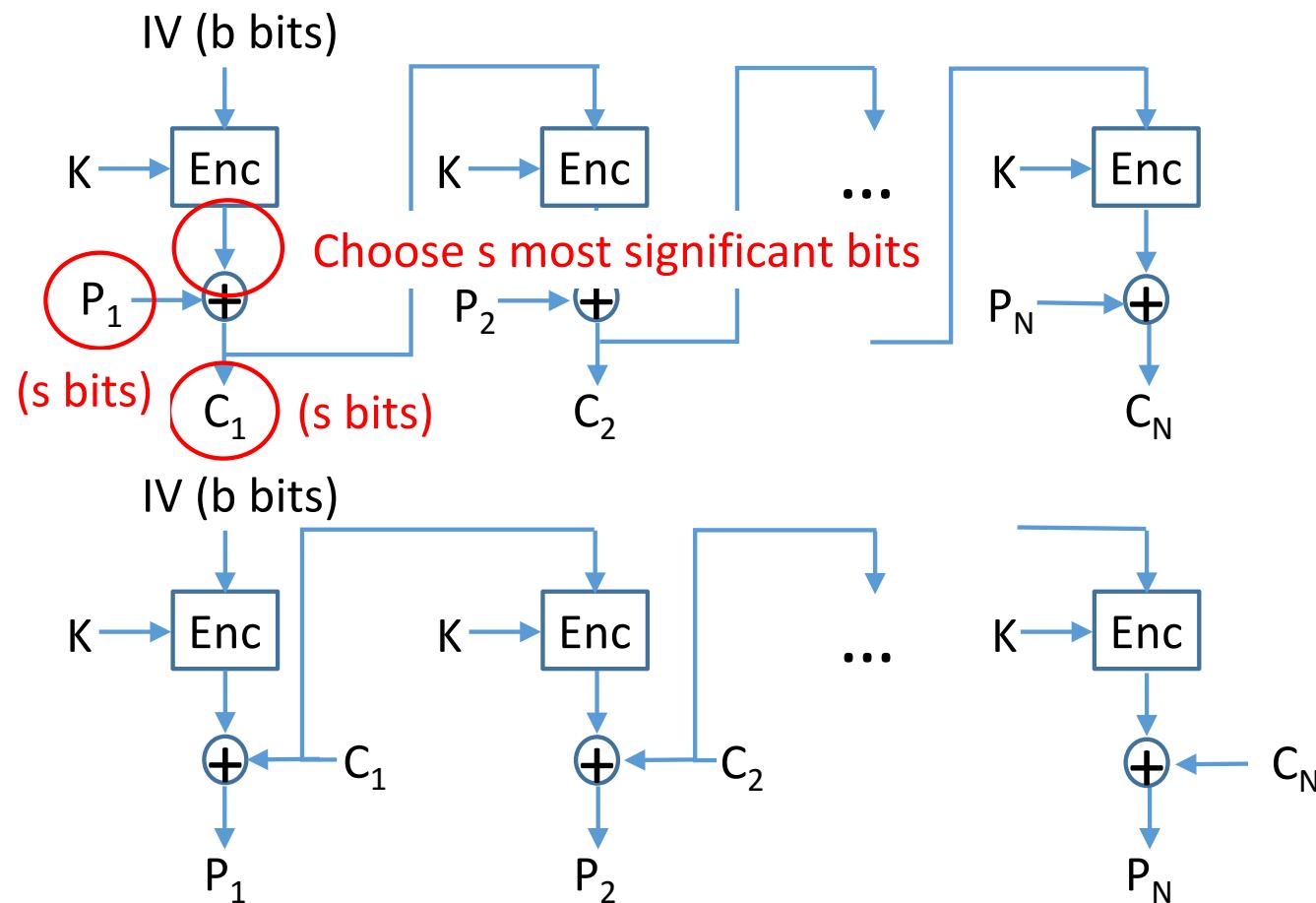




**CFB Encryption**  
 $C_i = P_i \oplus \text{Enc}(K, C_{i-1})$



**CFB Decryption**  
 $P_i = C_i \oplus \text{Enc}(K, C_{i-1})$



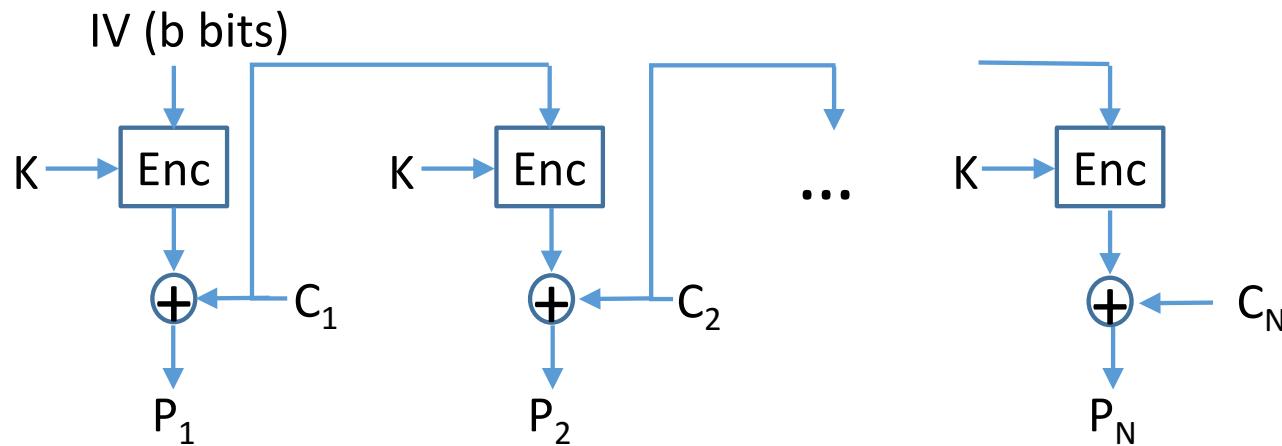
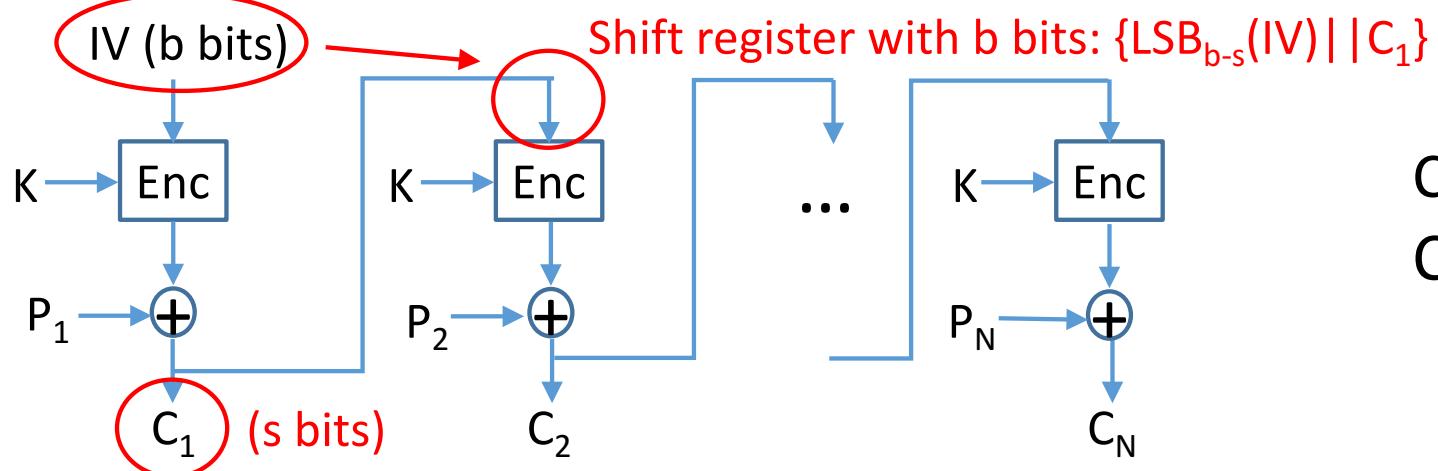
s-bit CFB mode

CFB Encryption

$$C_i = P_i \oplus \text{Enc}(K, C_{i-1})$$

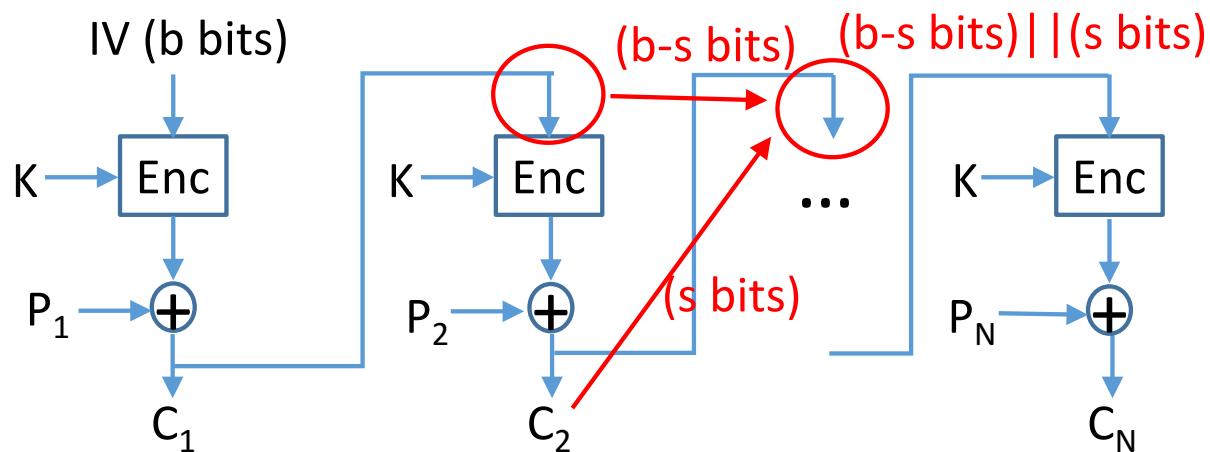
CFB Decryption

$$P_i = C_i \oplus \text{Enc}(K, C_{i-1})$$



CFB Encryption

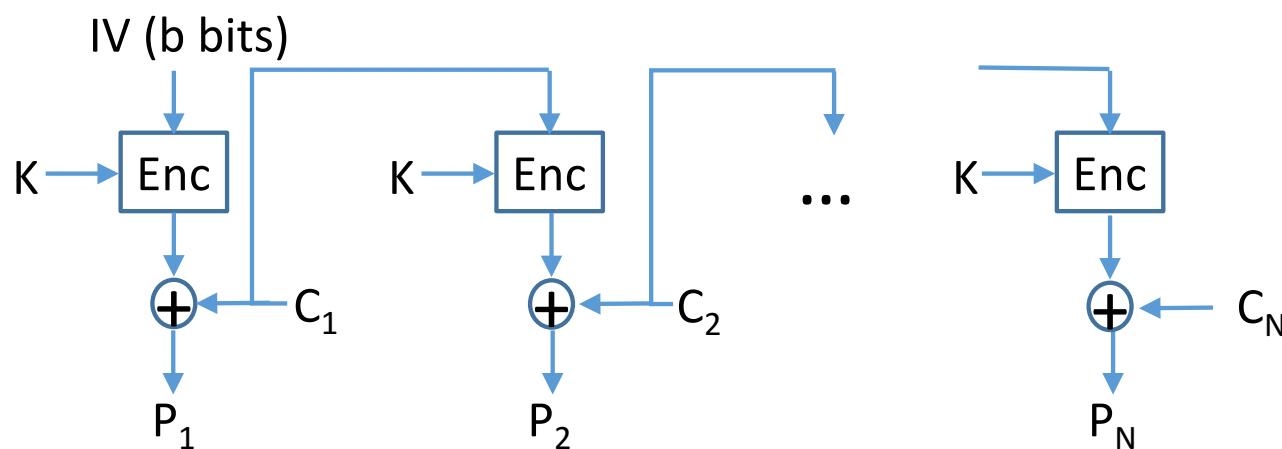
$$C_i = P_i \oplus \text{Enc}(K, C_{i-1})$$



s-bit CFB mode

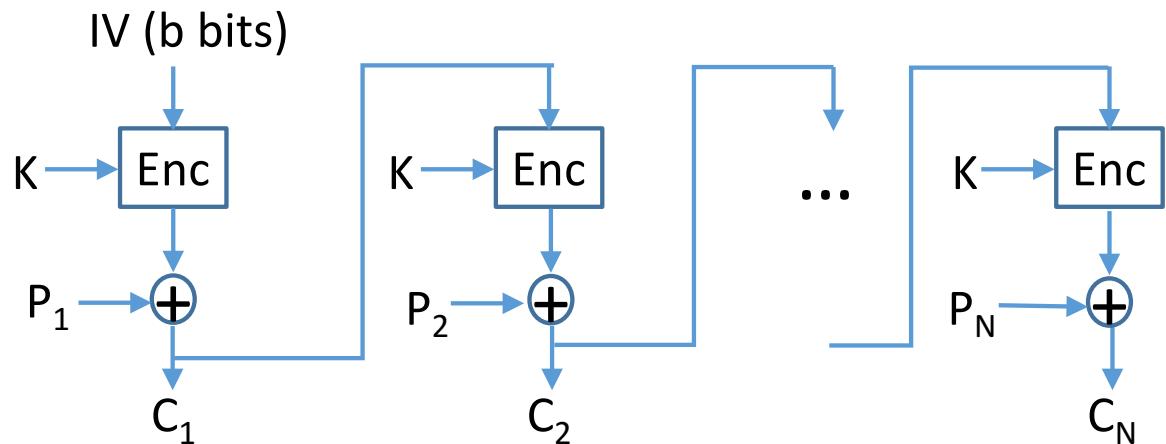
CFB Encryption

$$C_i = P_i \oplus \text{Enc}(K, C_{i-1})$$

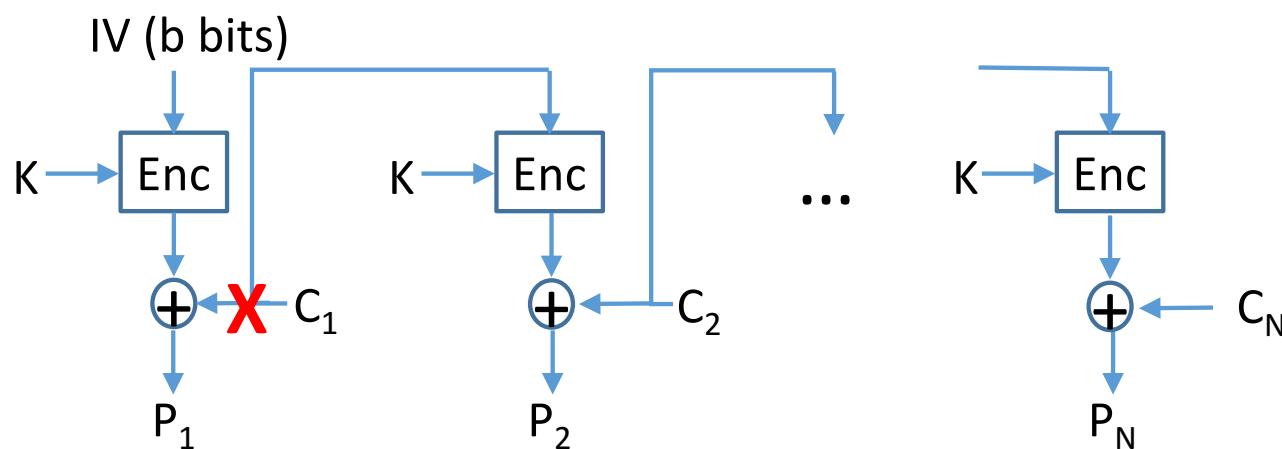


CFB Decryption

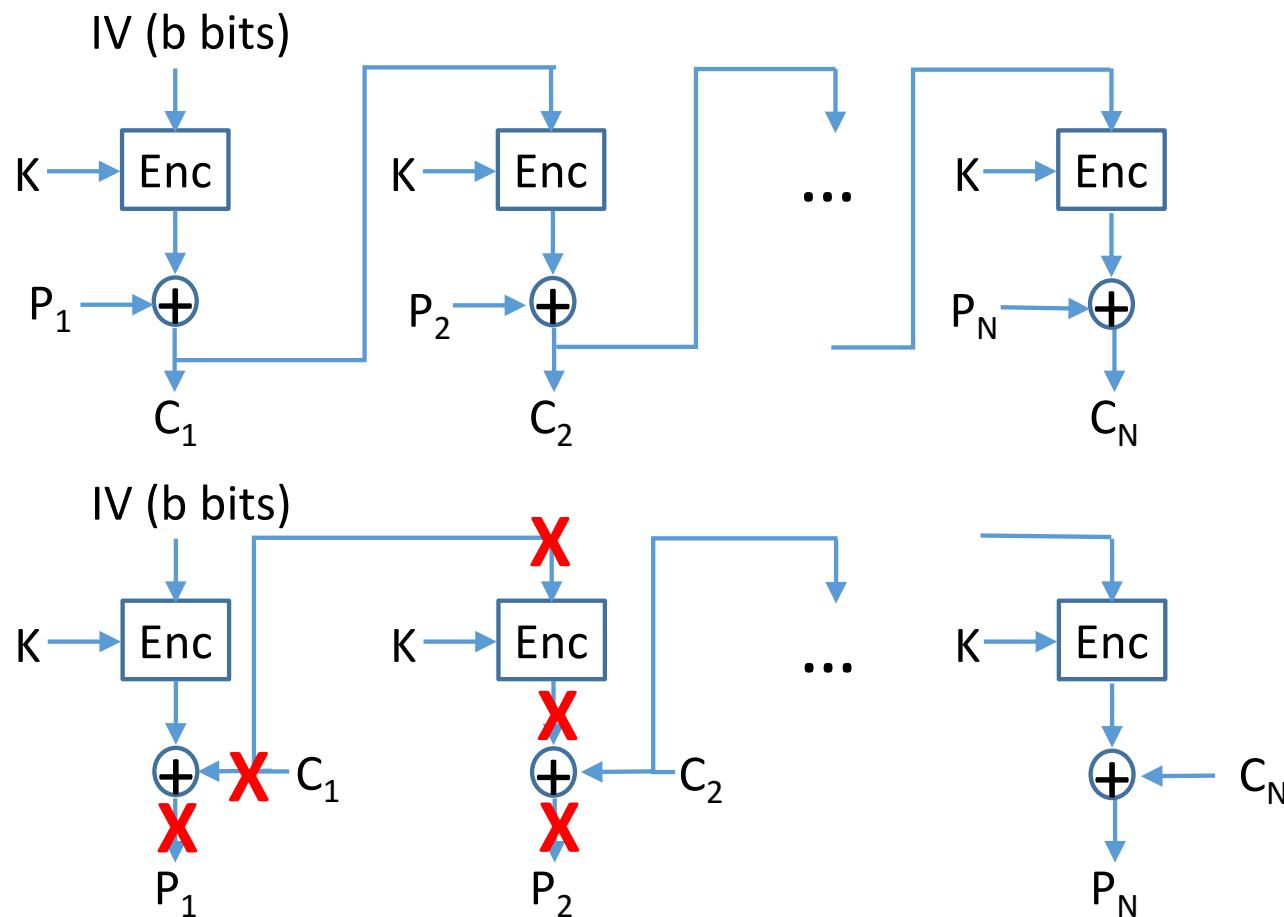
$$P_i = C_i \oplus \text{Enc}(K, C_{i-1})$$



**CFB Encryption**  
 $C_i = P_i \oplus \text{Enc}(K, C_{i-1})$



**CFB Decryption**  
 $P_i = C_i \oplus \text{Enc}(K, C_{i-1})$



**CFB Encryption**  
 $C_i = P_i \oplus \text{Enc}(K, C_{i-1})$

**CFB Decryption**  
 $P_i = C_i \oplus \text{Enc}(K, C_{i-1})$



## Output Feedback (OFB) Mode

Cipher function output is the feedback

Feedback is independent to  $P_i$  and  $C_i$

Encryption:  $C_i = P_i \oplus \text{Enc}(K, [C_{i-1} \oplus P_{i-1}])$

Decryption:  $P_i = C_i \oplus \text{Enc}(K, [C_{i-1} \oplus P_{i-1}])$

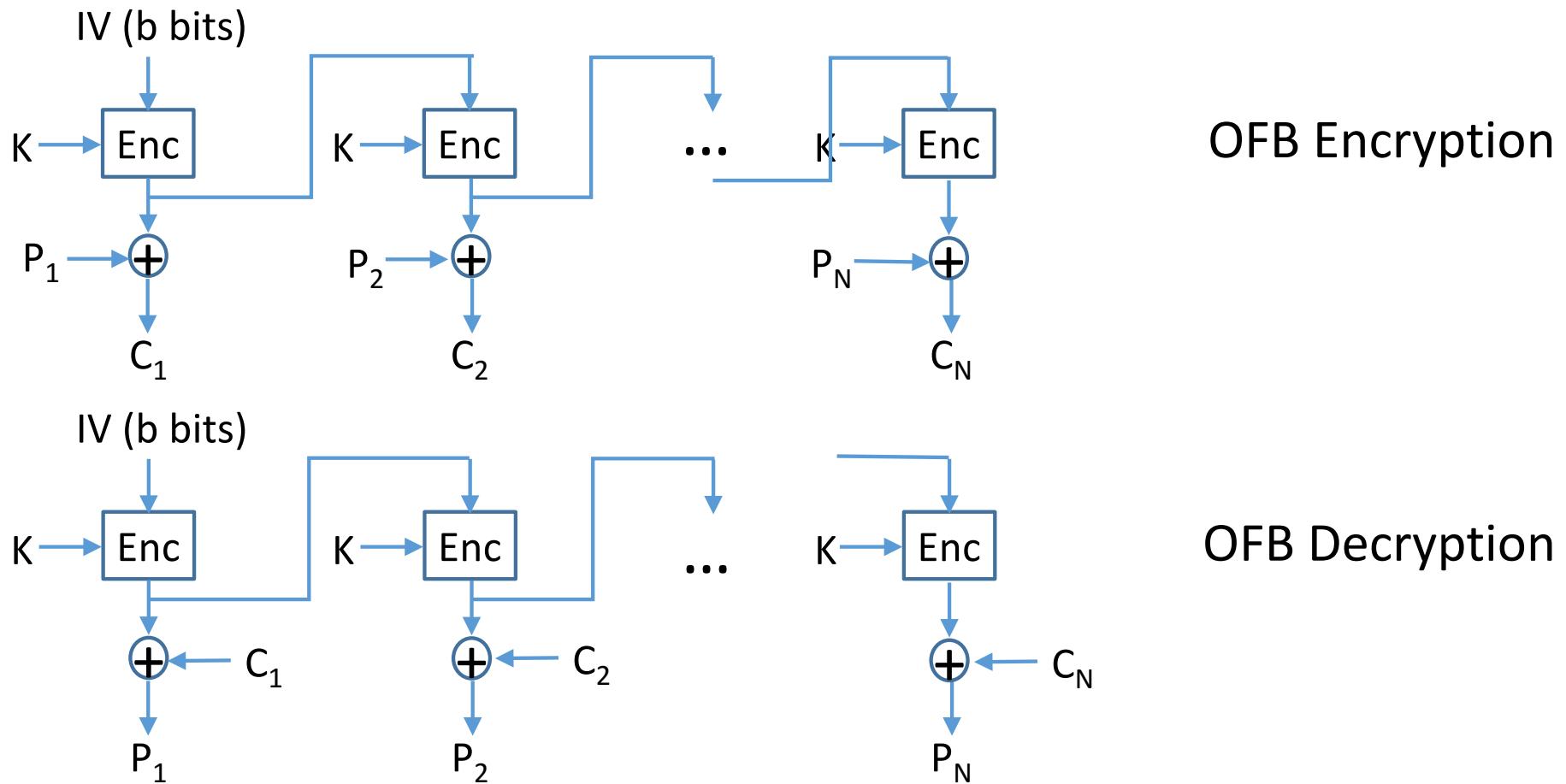
## Output Feedback (OFB) Mode

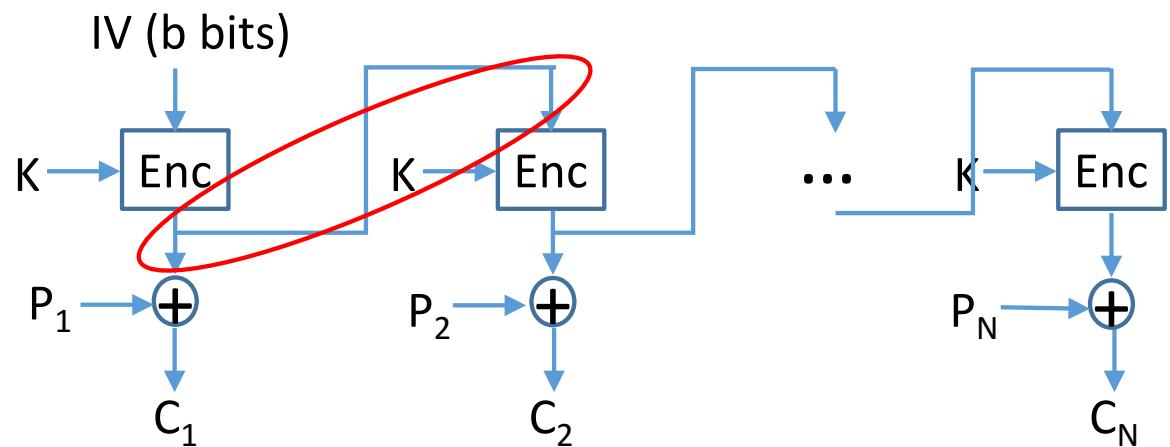
Cipher function output is the feedback

Feedback is independent to  $P_i$  and  $C_i$

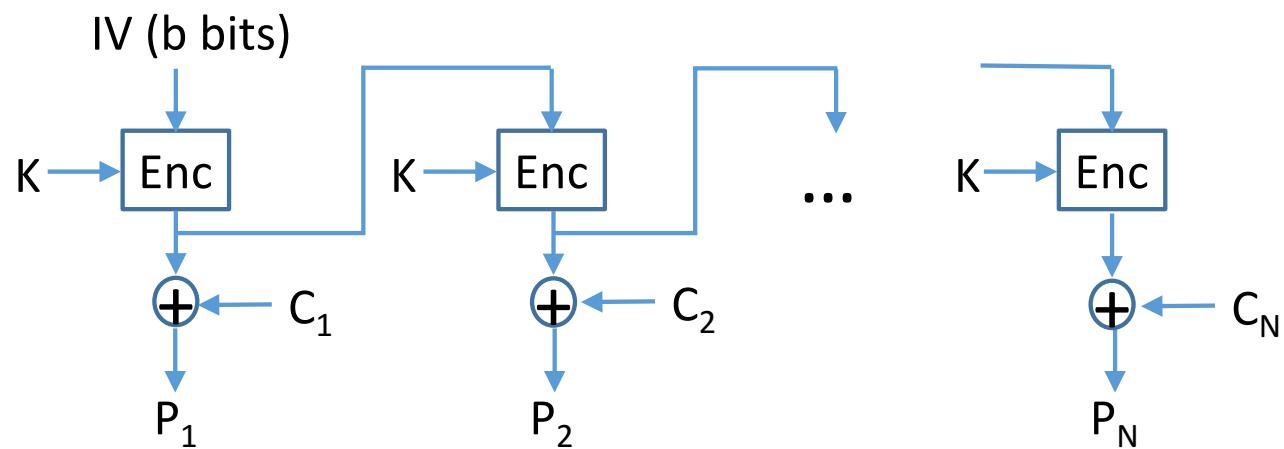
Encryption:  $C_i = P_i \oplus \text{Enc}(K, [C_{i-1} \oplus P_{i-1}])$

Decryption:  $P_i = C_i \oplus \text{Enc}(K, [C_{i-1} \oplus P_{i-1}])$

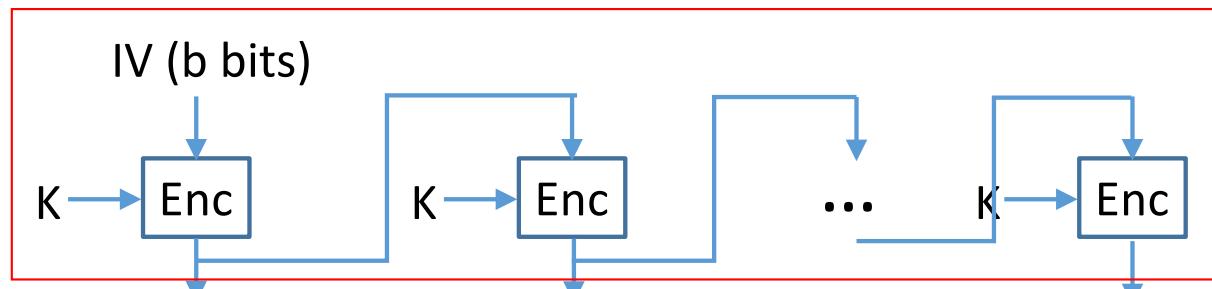




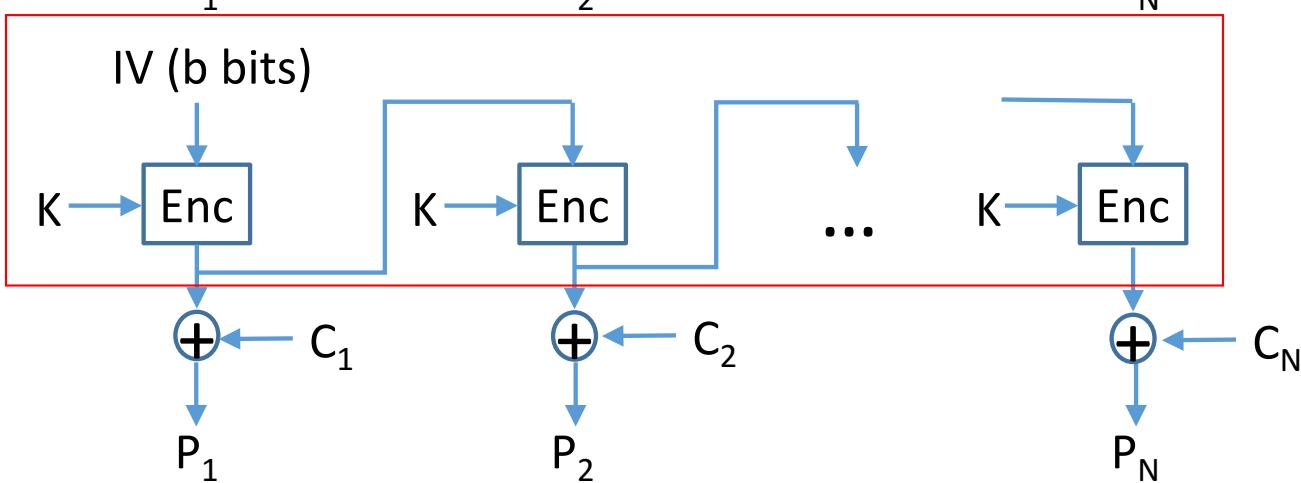
OFB Encryption



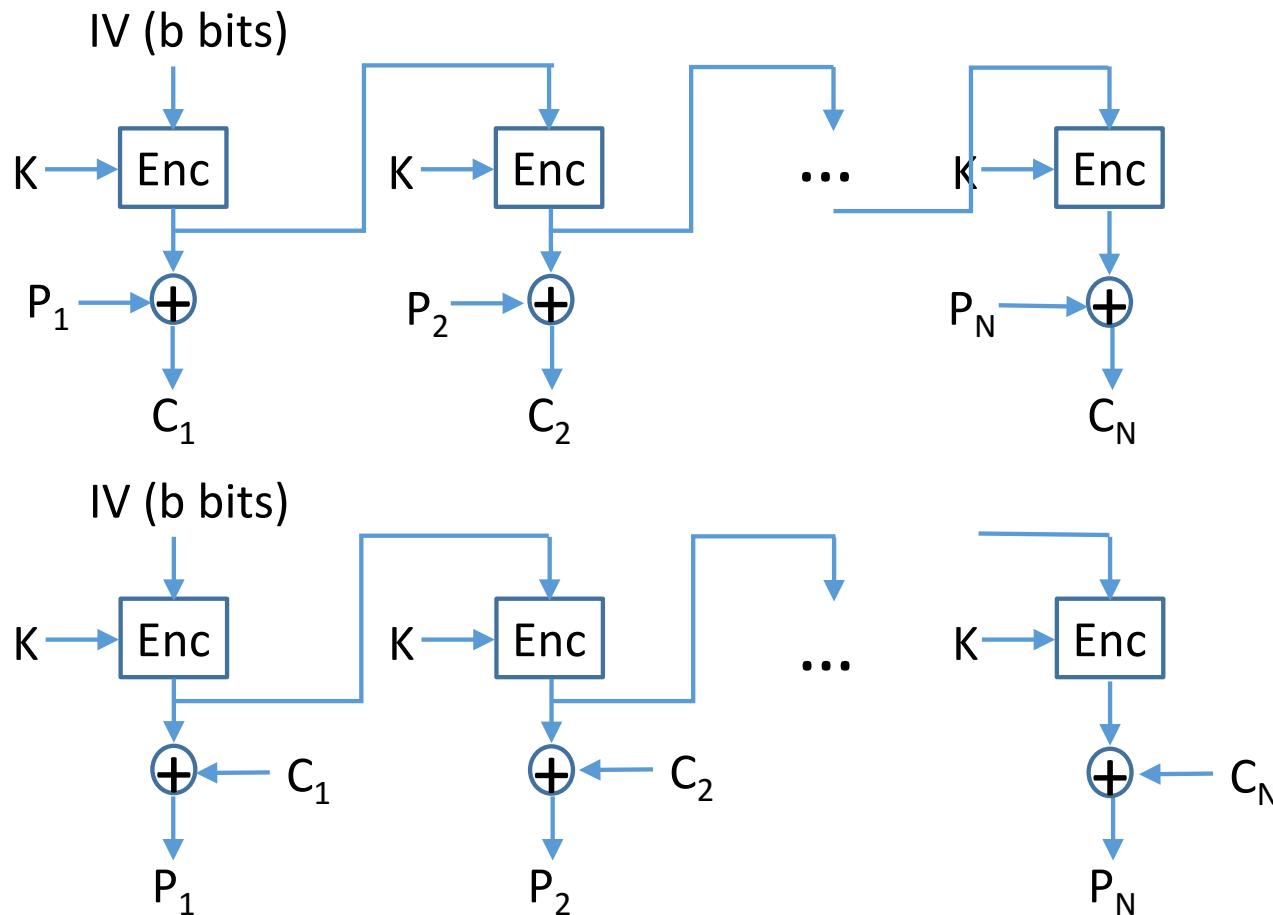
OFB Decryption



OFB Encryption



OFB Decryption

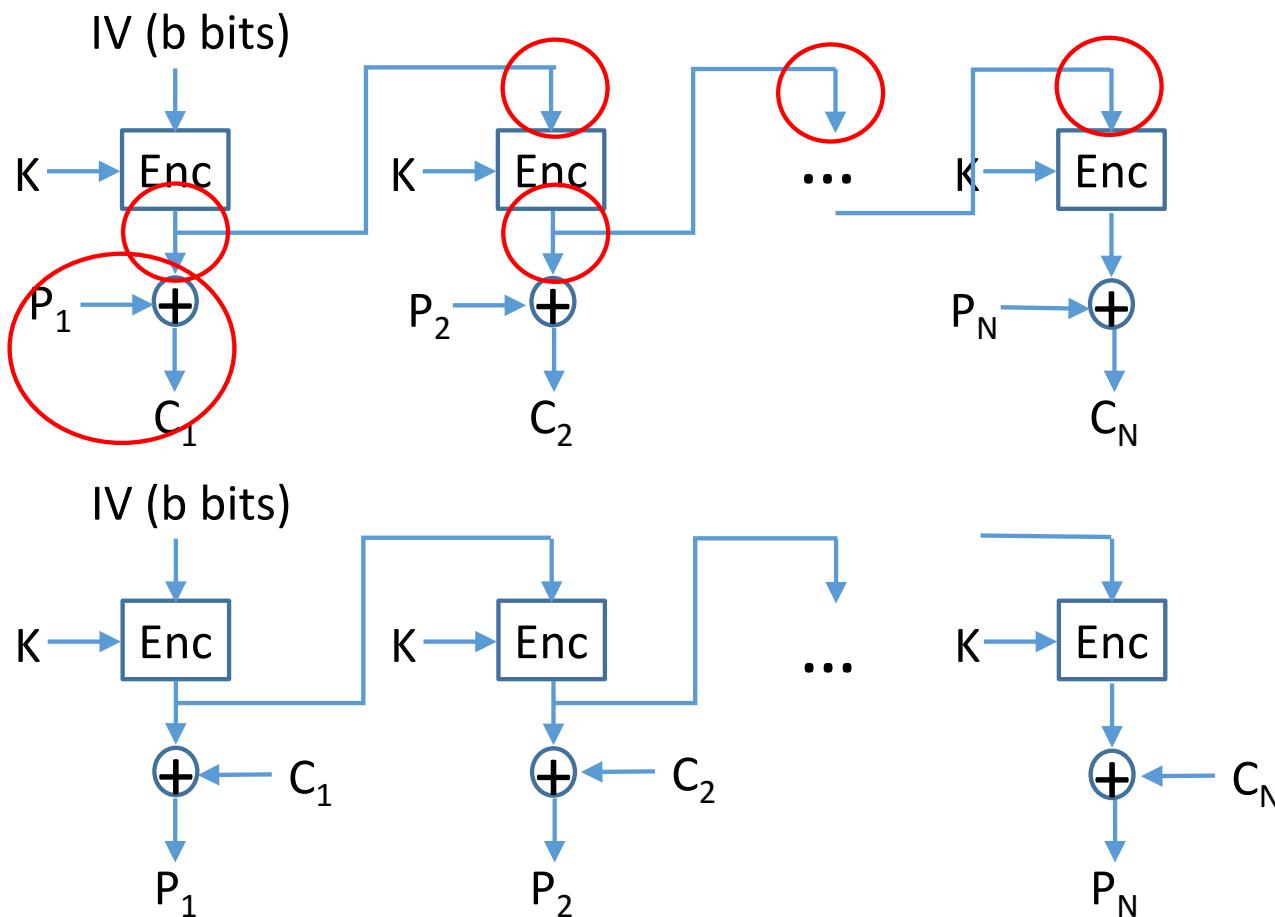


OFB Encryption

$$C_i = P_i \oplus \text{Enc}(K, [C_{i-1} \oplus P_{i-1}])$$

OFB Decryption

$$P_i = C_i \oplus \text{Enc}(K, [C_{i-1} \oplus P_{i-1}])$$



OFB Encryption

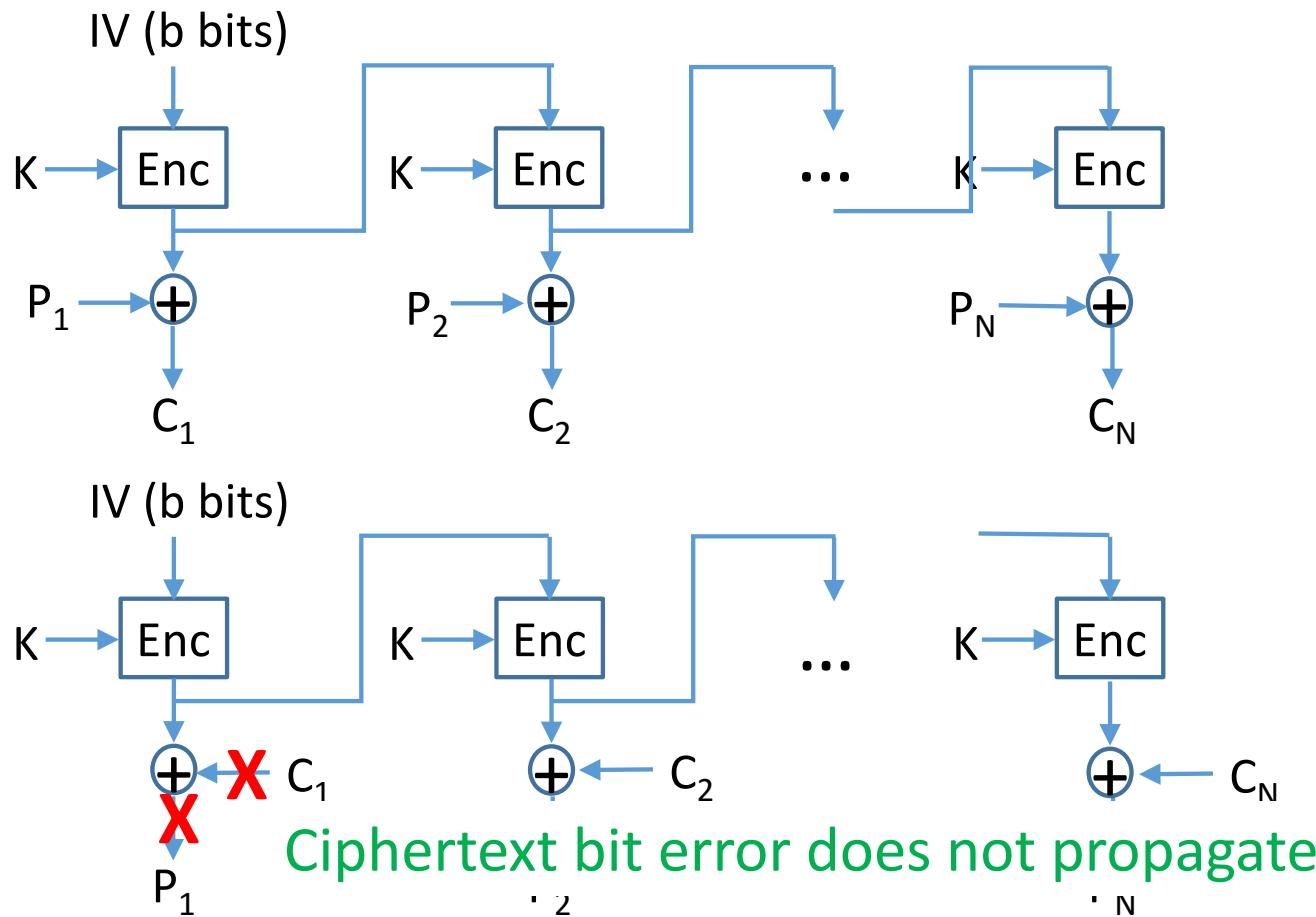
$$C_i = P_i \oplus \text{Enc}(K, [C_{i-1} \oplus P_{i-1}])$$

Because

$$X \oplus Y = Z \rightarrow X = Y \oplus Z$$

OFB Decryption

$$P_i = C_i \oplus \text{Enc}(K, [C_{i-1} \oplus P_{i-1}])$$

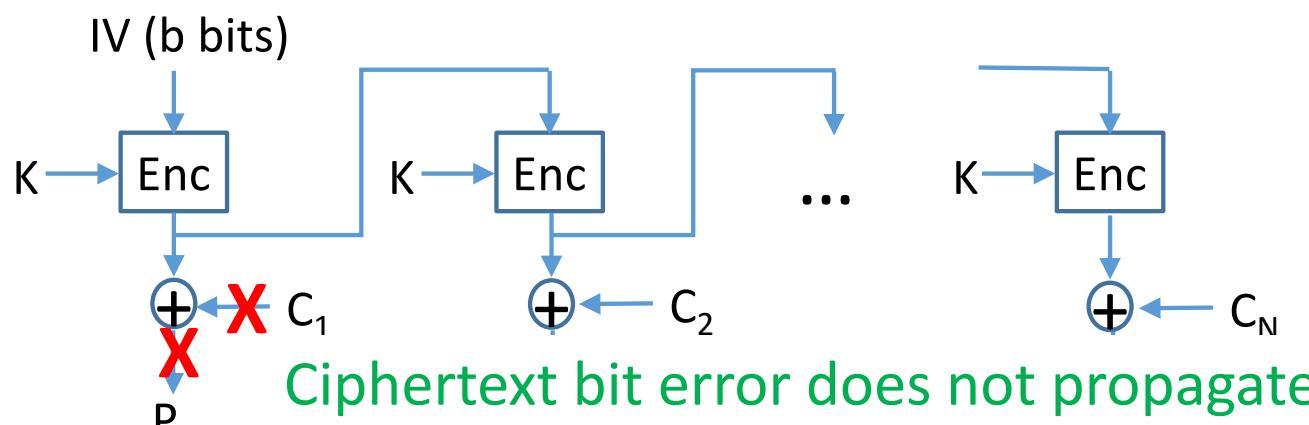
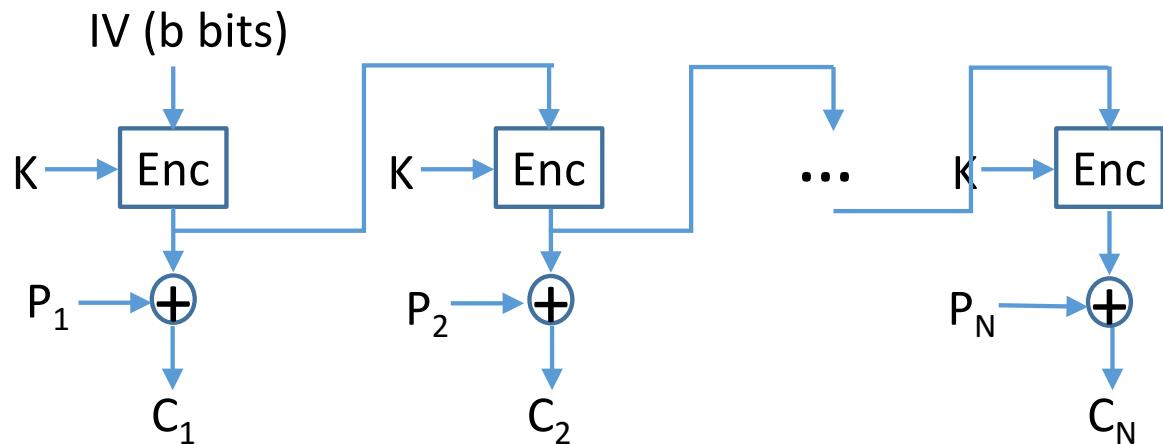


OFB Encryption

$$C_i = P_i \oplus \text{Enc}(K, [C_{i-1} \oplus P_{i-1}])$$

OFB Decryption

$$P_i = C_i \oplus \text{Enc}(K, [C_{i-1} \oplus P_{i-1}])$$



Ciphertext bit error does not propagate

Vulnerable to attacker's message stream modification

OFB Encryption

$$C_i = P_i \oplus \text{Enc}(K, [C_{i-1} \oplus P_{i-1}])$$

OFB Decryption

$$P_i = C_i \oplus \text{Enc}(K, [C_{i-1} \oplus P_{i-1}])$$



## Counter (CTR) Mode

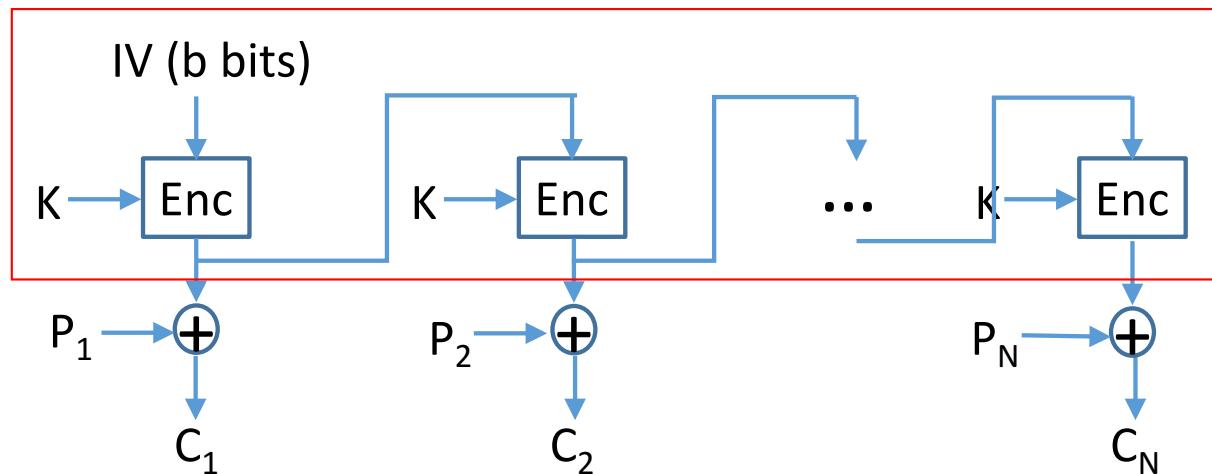
No explicit chaining/feedback

Counter for pseudo-random generation

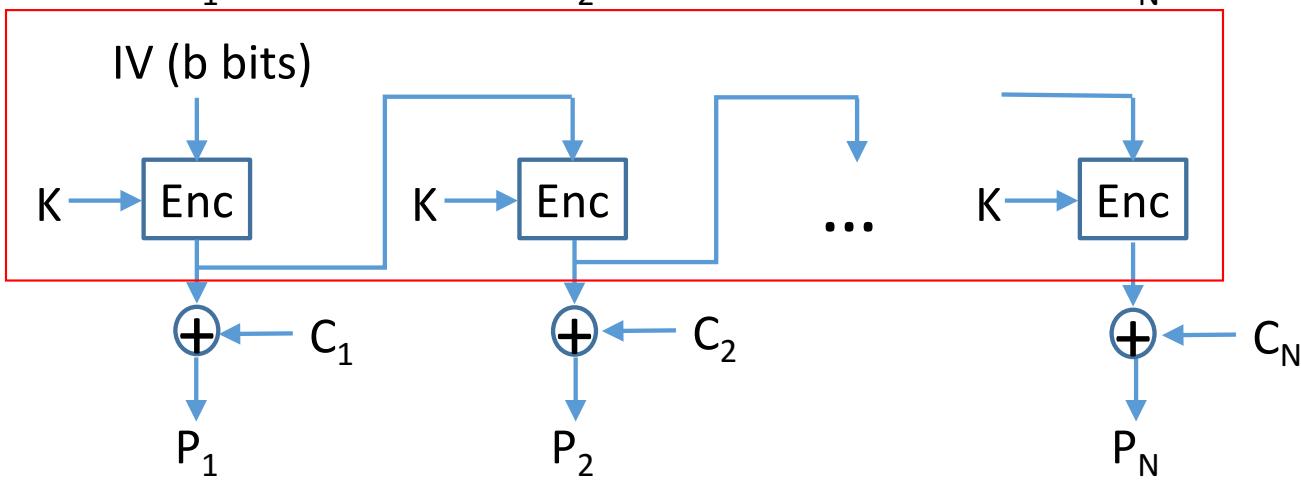
The initial counter ( $T_1$ ) must be nonce

Encryption:  $C_i = P_i \oplus \text{Enc}(K, T_i)$

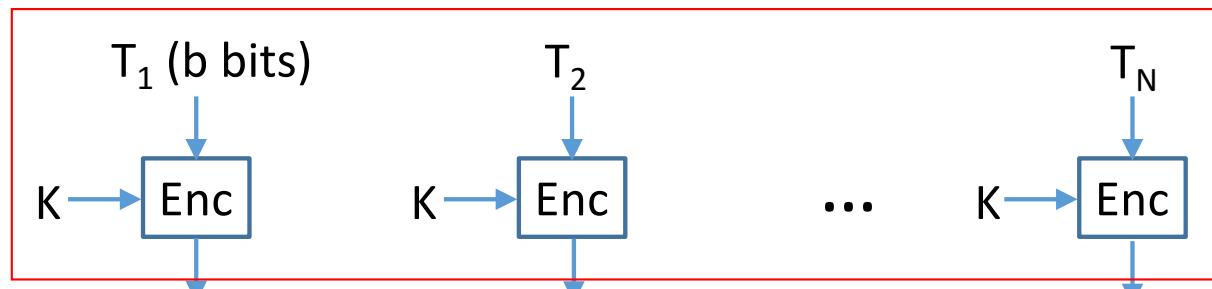
Decryption:  $P_i = C_i \oplus \text{Enc}(K, T_i)$



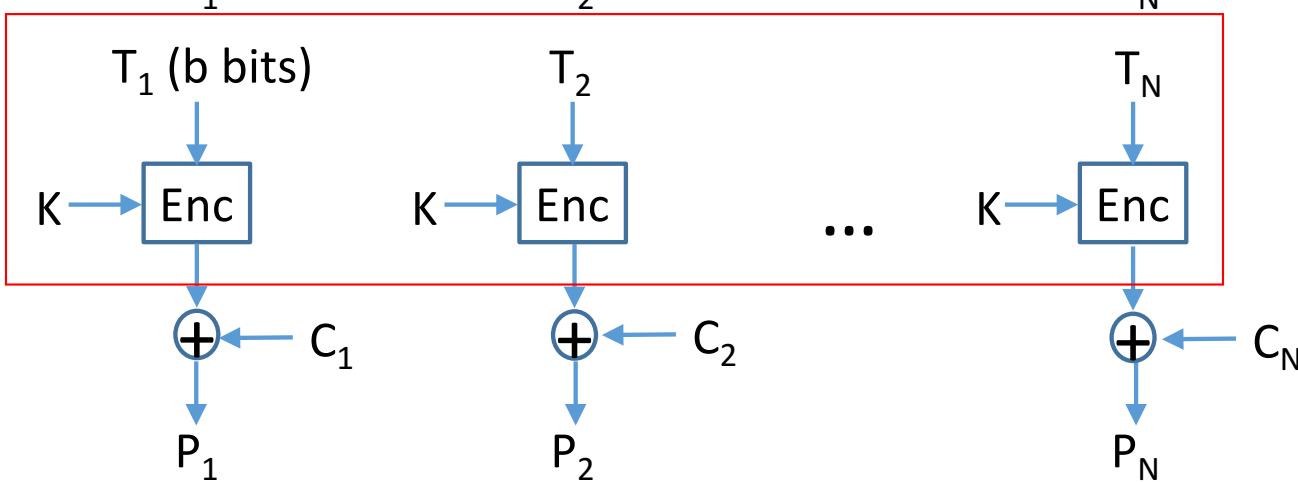
**OFB Encryption**



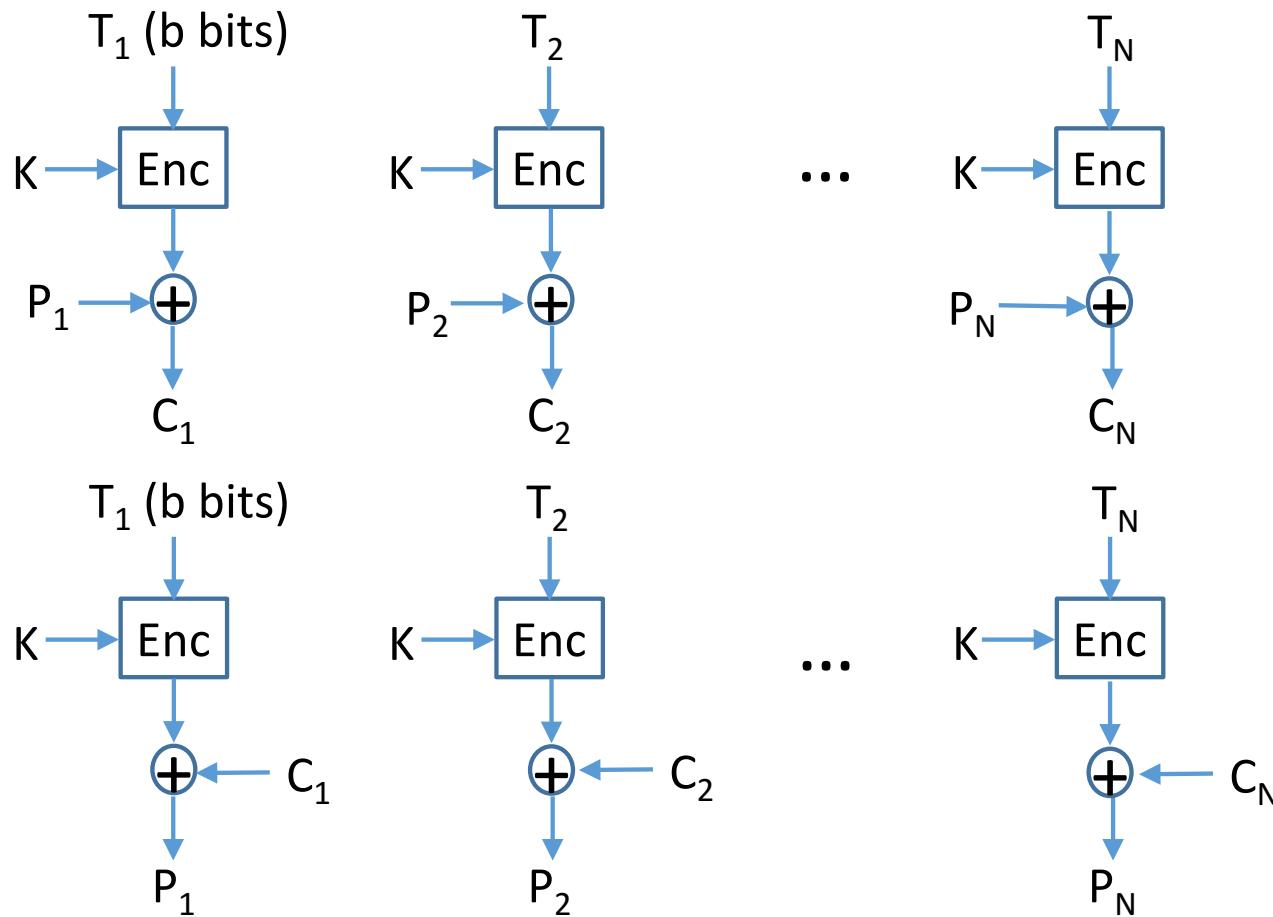
**OFB Decryption**



CTR Encryption

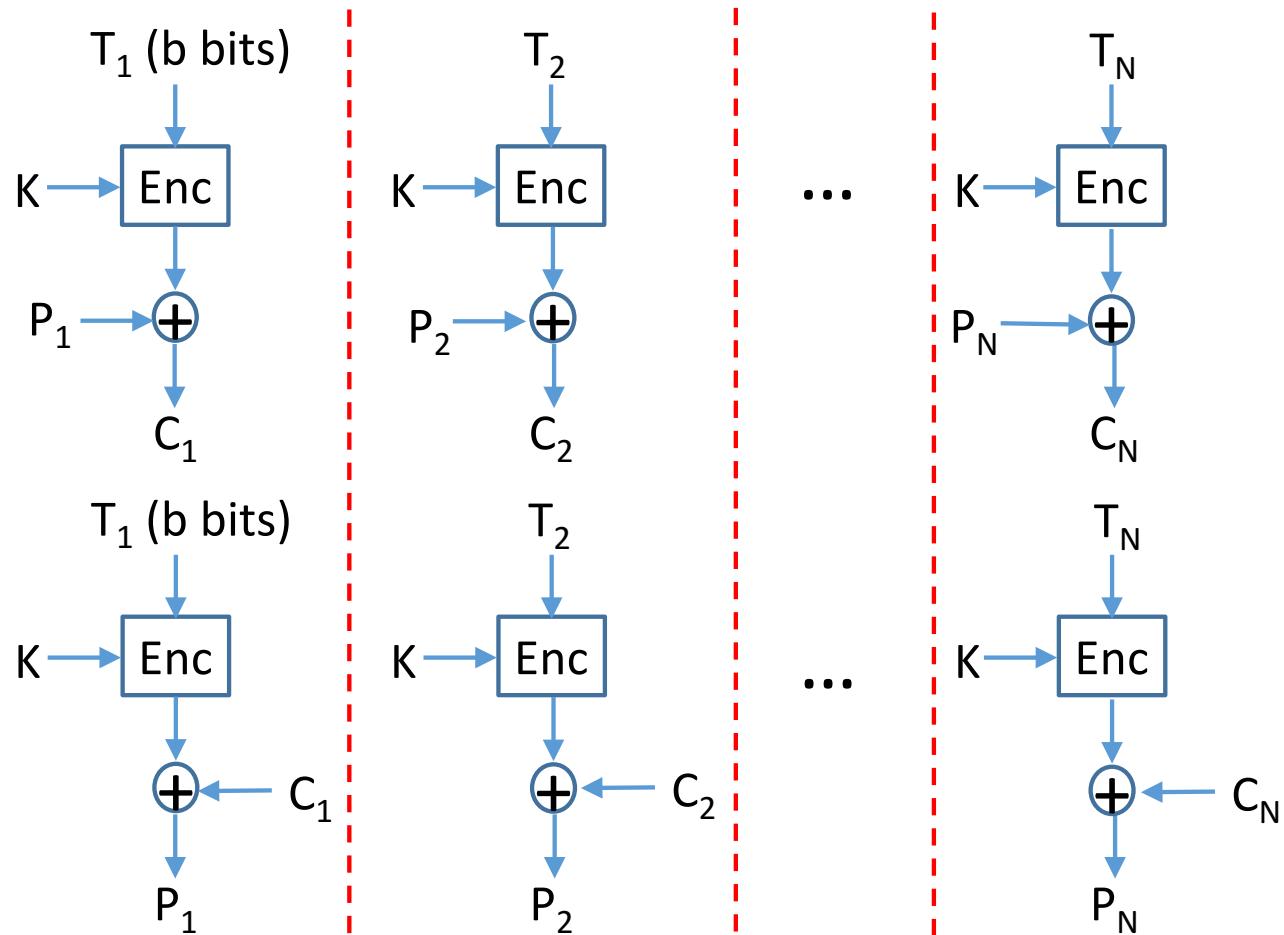


CTR Decryption



CTR Encryption  
 $C_i = P_i \oplus \text{Enc}(K, T_i)$

CTR Decryption  
 $P_i = C_i \oplus \text{Dec}(K, T_i)$

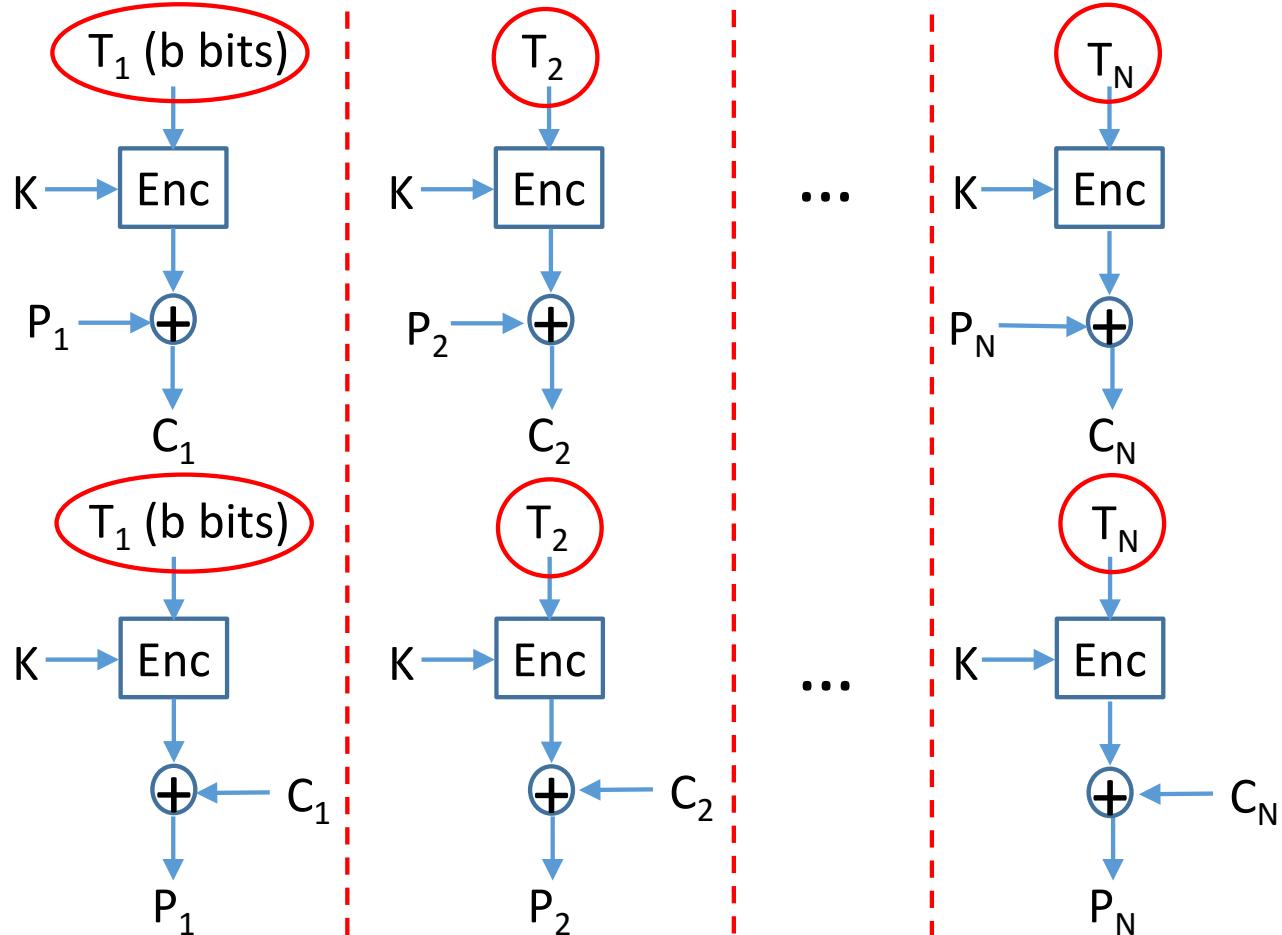


CTR Encryption

$$C_i = P_i \oplus \text{Enc}(K, T_i)$$

CTR Decryption

$$P_i = C_i \oplus \text{Enc}(K, T_i)$$



CTR Encryption

$$C_i = P_i \oplus \text{Enc}(K, T_i)$$

CTR Decryption

$$P_i = C_i \oplus \text{Enc}(K, T_i)$$

## **CTR Mode Use**

Standard requires that the counter  
is initialized to different values every  
time it is used (every plaintext has  
Different  $T_1$ )

Used in Asynchronous Transfer Mode  
network security and IPSec (IP security)

## **Block Cipher Modes Feedback**

ECB, CBC, CFB, OFB, CTR

Except for ECB, chaining and  
dependence across blocks

For CTR, the CTR value is dependent  
(implicit feedback within counter)

