Tutorial 3

CSCI361 – Computer Security

Sionggo Japit
sjapit@uow.edu.au

TUTORIAL 2

Different Mode of Block Operations

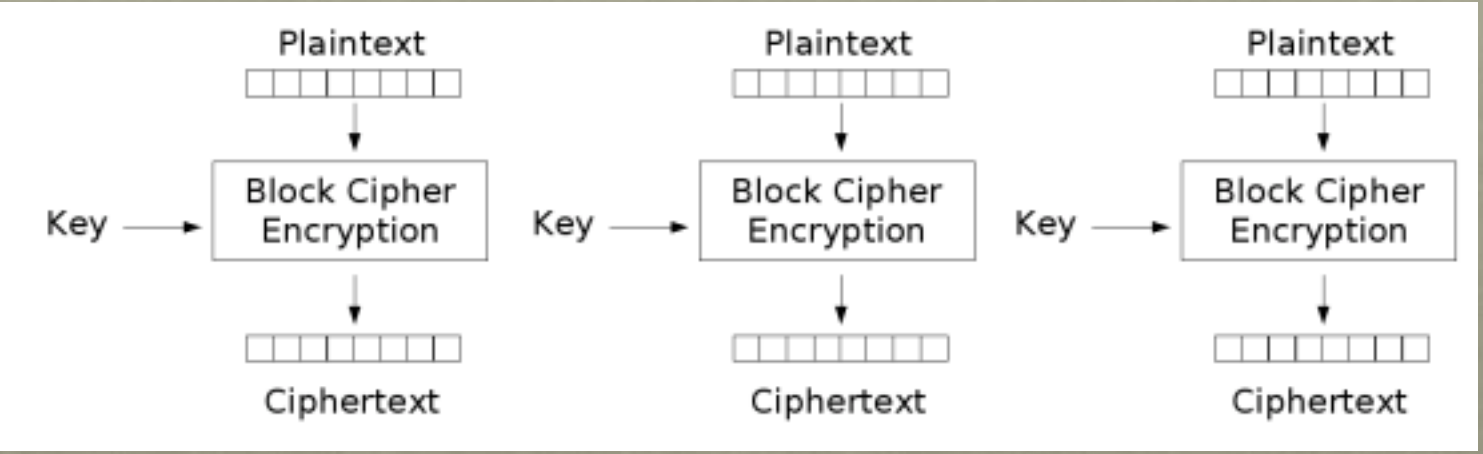
DIFFERENT MODES OF OPERATION

Different modes of block operations:

1. Electronic codebook mode (ECB)
2. Cipher block chaining mode (CBC)
3. Cipher feedback mode (CFB)
4. Output feedback mode (OFB)
5. Counter mode (CTR)

ELECTRONIC CODEBOOK MODE (ECB)

- ECB mode corresponds to the usual use of a block cipher:
  - Given a sequence $x_1x_2\ldots$of 64-bit plaintext blocks, each x_i is encrypted with the same key K, producing a string of ciphertext blocks, $y_1y_2\ldots$.



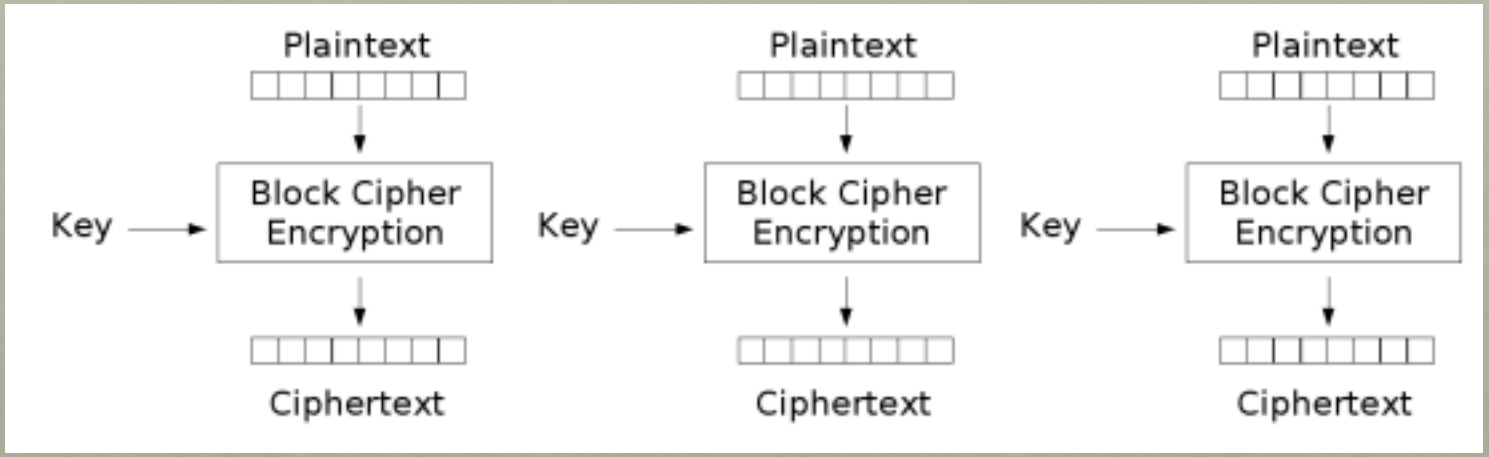
Electronic Codebook Mode (ECB)

- Parallel Encryption possible
- Same block will result in having the same ciphertext
  - Dictionary attack, leaks some information
- Subjected to modification of blocks
  - “Reuse” ciphertext from other messages

ELECTRONIC CODEBOOK MODE (ECB)

- The simplest mode of the encryption modes.
- The plaintext is divided into blocks, and each block of plaintext is encrypted separately using the same key.
- For a given key, there is a unique ciphertext for every b-bit block of plaintext.
- For a message longer than *b* bits, the message is broken into b-bit blocks, padding the last block if necessary.
- Decryption is performed one block at a time, always using the same key.
- Parallel Encryption possible.

ELECTRONIC CODEBOOK MODE (ECB)



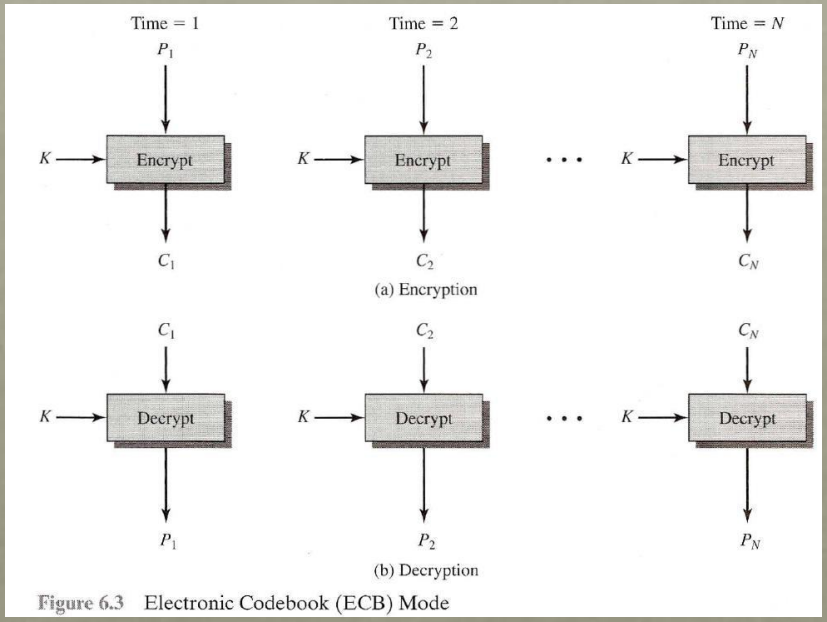


Figure 6.3 Electronic Codebook (ECB) Mode

$$c_i = E_k(m_i)$$

$$m_i = D_k(c_i)$$

ELECTRONIC CODEBOOK MODE (ECB)

- The ECB method is ideal for a short amount of data, such as encryption key.
- Disadvantage:
  - Identical plaintext blocks are encrypted into identical ciphertext block, thus, it does not hide data patterns well.

ELECTRONIC CODEBOOK MODE (ECB)

- Let us consider a 3-bit block cipher with the following mapping:

Input	000	001	010	011	100	101	110	111
Output	111	110	011	100	001	000	101	010

ECB

Plaintext	101	101	110	010
Ciphertext	000	000	101	011

CIPHER BLOCK CHAINING MODE (CBC)

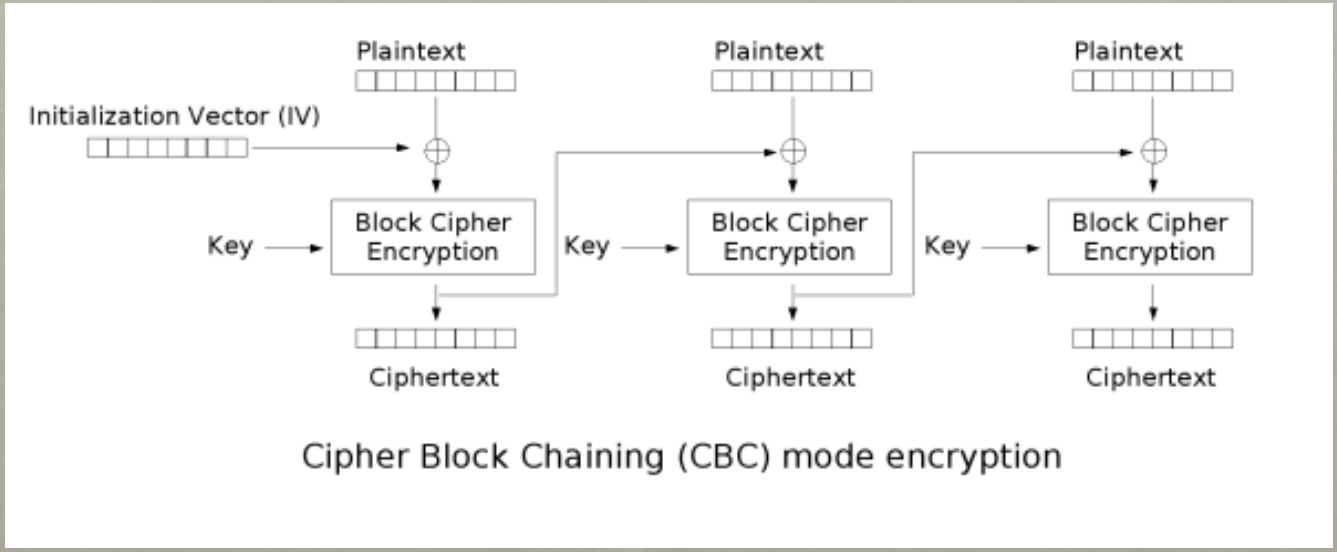
- Input to the encryption algorithm is the XOR of the current plaintext block and the preceding ciphertext block; this produces the chaining effect, i.e., each ciphertext block is dependent on all plaintext blocks processed up to that point.
- For the first block, initial vector (IV) is used. IV is not secret.
- The same key is used for each block.
- Parallel encryption is not possible.

CIPHER BLOCK CHAINING MODE (CBC)

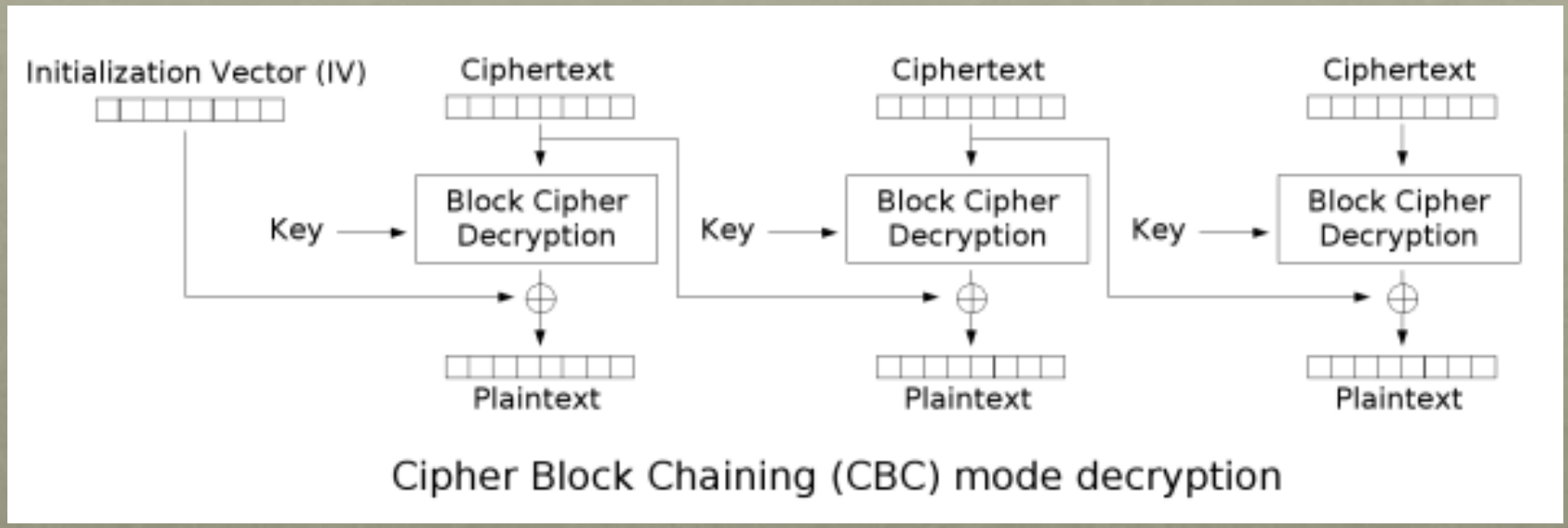
- The fact that each ciphertext block is dependent on all plaintext blocks processed so far, identical plaintext blocks will be encrypted to a different ciphertext block. In other words, repeating patterns of *b* bits are not exposed, an advantage over ECB.

$$C_i = E(K, [C_{i-1} \oplus P_i), C_0 = IV$$

CIPHER BLOCK CHAINING MODE (CBC)



Cipher Block Chaining (CBC) mode encryption



Cipher Block Chaining (CBC) mode decryption

CIPHER BLOCK CHAINING MODE (CBC)

- To decrypt, each cipher block is passed through the decryption algorithm.
- The result is XORed with the preceding ciphertext block to produce the plaintext block.

$$P_i = D(K, C_i) \oplus C_{i-1}, C_0 = IV$$

- A plaintext can be recovered from just two adjacent blocks of the ciphertext, thus, decryption can be parallelized.

CIPHER BLOCK CHAINING MODE (CBC)

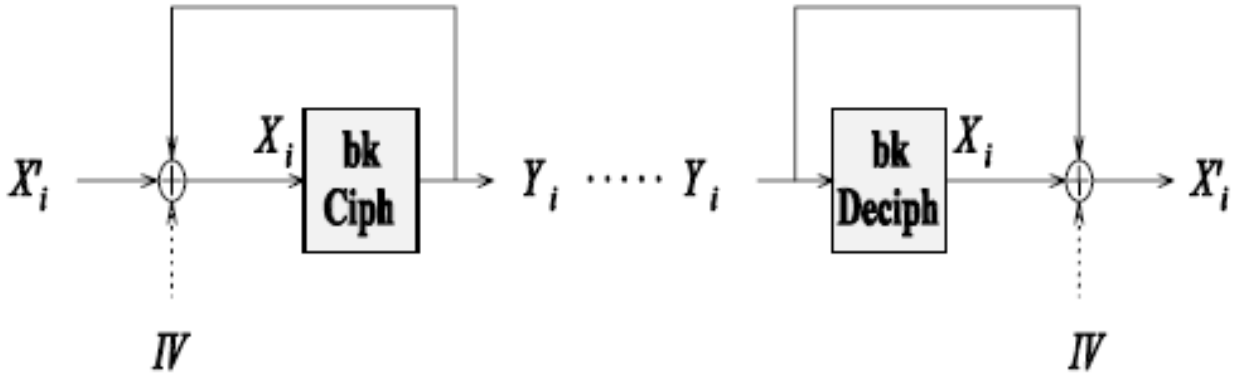
- Disadvantage:
  - Due to the chaining, the encryption is sequential, i.e., the encryption process cannot be parallelized.
  - A one-bit change in a plaintext affects all following ciphertext blocks.

CIPHER BLOCK CHAINING MODE (CBC)

Input	000	001	010	011	100	101	110	111
Output	111	110	011	100	001	000	101	010

$XOR_0 = Plaintext_0 \oplus IV$

$XOR_i = Plaintext_i \oplus Ciphertext_{i-1}$



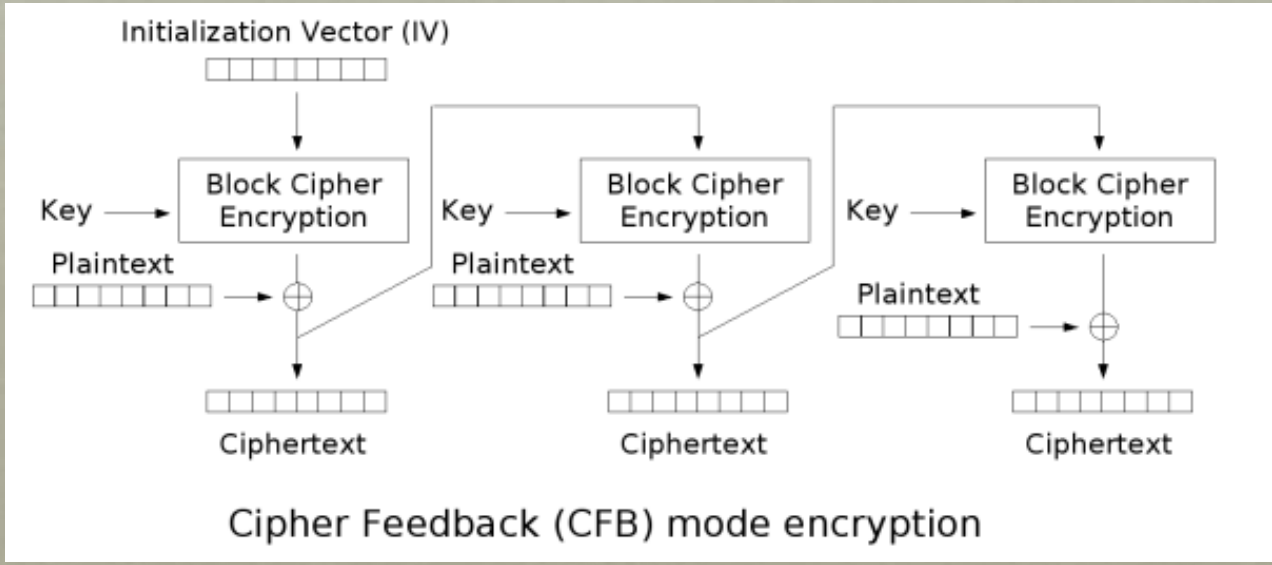
CBC
IV=111

Plaintext	101	101	110	010
XOR	010	110	011	110
Ciphertext	011	101	100	101

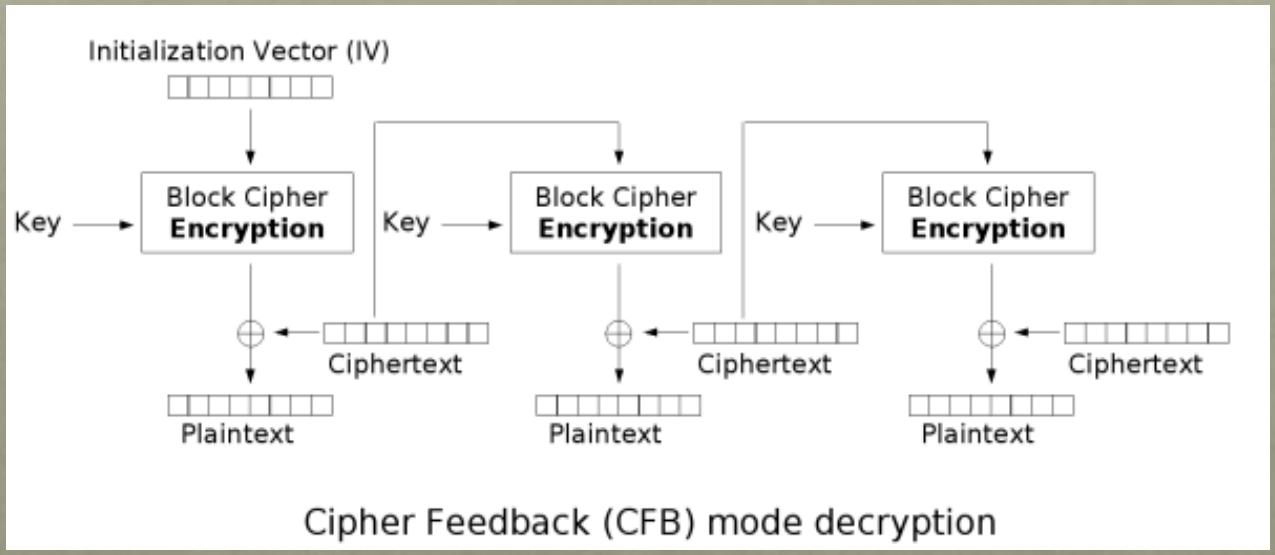
Cipher Feedback Mode (CFB)

- A close relative of CBC.
- Similarly to CBC, the units of plaintext are chained together, so that the ciphertext of any plaintext unit is a function of all the preceding plaintext.
- Cipher Feedback (CFB) makes a block cipher into a self-synchronizing stream cipher.

CIPHER FEEDBACK MODE (CFB)



Cipher Feedback (CFB) mode encryption



Cipher Feedback (CFB) mode decryption

CIPHER FEEDBACK MODE (CFB)

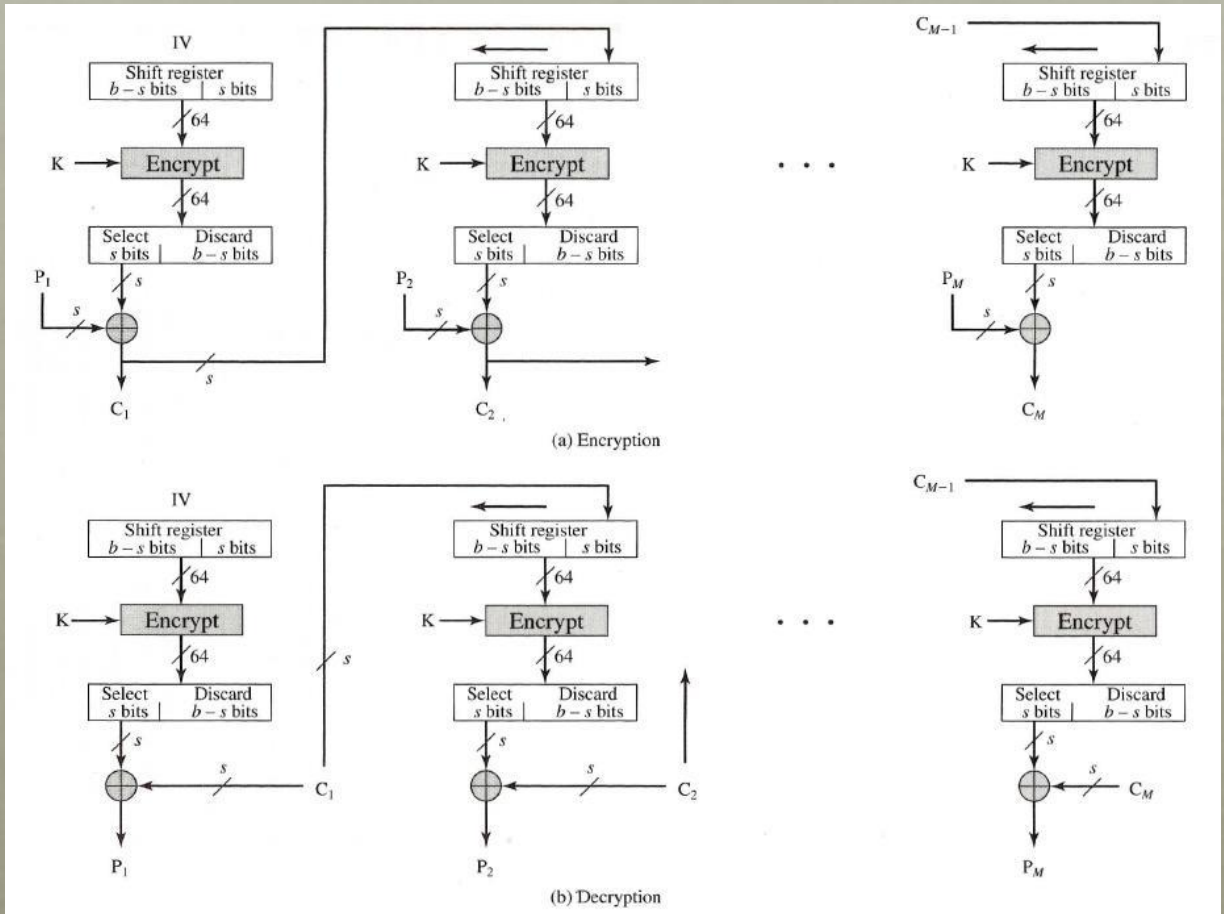


Figure 6.5 *s*-bit Cipher Feedback (CFB) Mode

CIPHER FEEDBACK MODE (CFB)

- To encrypt, the input to the encryption function is a *b*-bit shift register that is initially set to some initialization vector (IV).
- The leftmost *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.
- The contents of the shift register are shifted left by *s* bits and C_1 is placed in the rightmost *s* bits of the shift register.

CIPHER FEEDBACK MODE (CFB)

- To decrypt, 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.
- The formula to encrypt and decrypt are as follows:

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

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

$$C_0 = IV$$

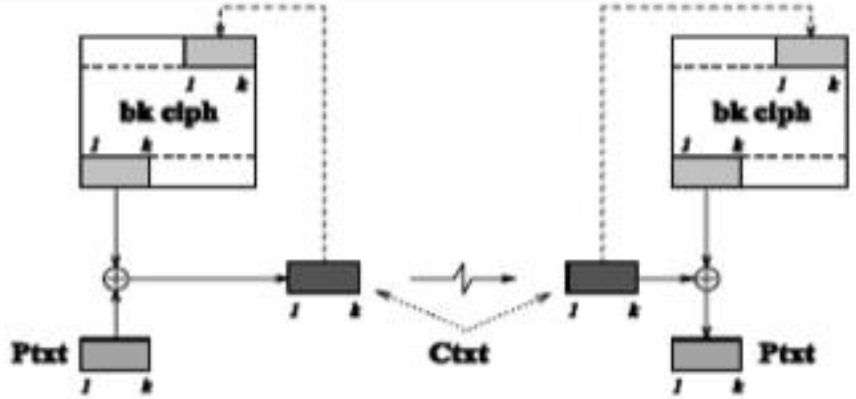
CIPHER FEEDBACK MODE (CFB)

- Similar to CBC mode, changes in the plaintext propagate in the ciphertext, and encryption cannot be parallelized, but decryption can.
- Advantage:
  - Same encryption function is used to encrypt and decrypt.
  - The message (plaintext) does not need to be padded to a multiple of the cipher block size.

CIPHER FEEDBACK MODE (CFB)

Input	000	001	010	011	100	101	110	111
Output	111	110	011	100	001	000	101	010

2-bit CFB
IV=111

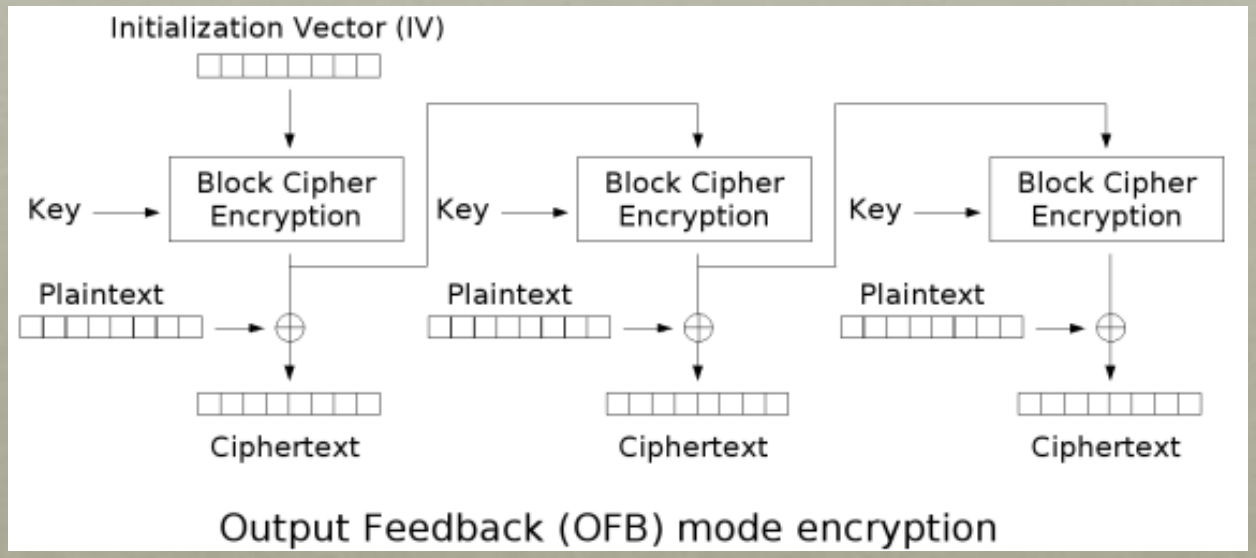


Plaintext	10	11	01	11	00	10
Input	111	**111**	**110**	**011**	**101**	**100**
E(Input)	**010**	**010**	**101**	**100**	**000**	**001**
Ciphertext	11	10	11	01	00	10

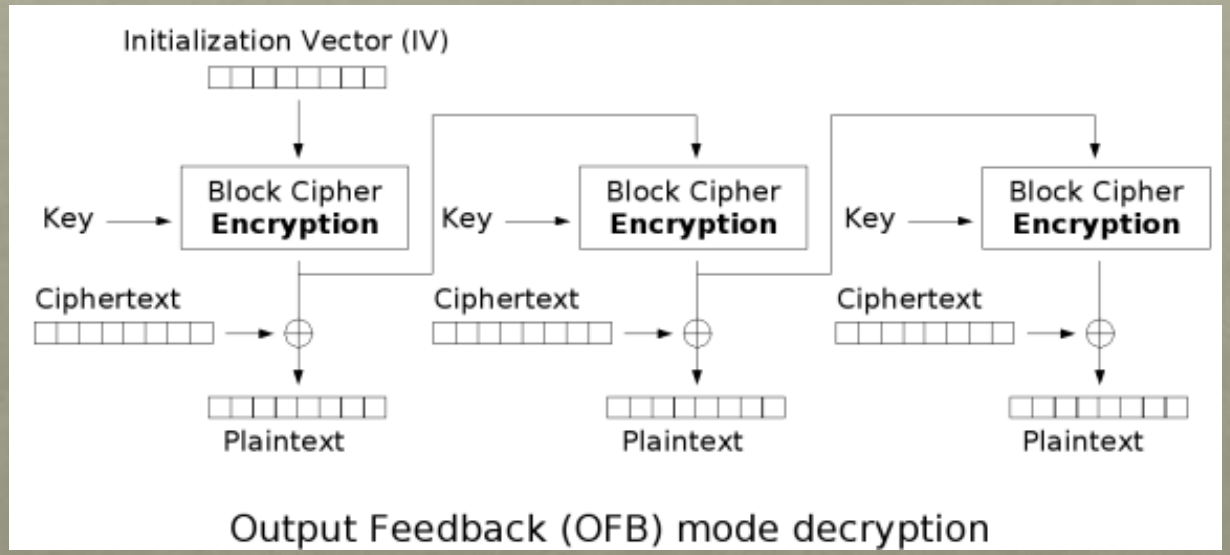
OUTPUT FEEDBACK MODE (OFB)

- Similar to CFB, except that the output of the encryption function is fed back to the shift register instead of the ciphertext unit.
- This give the advantage of not propagating bit error.
- Disadvantage:
  - More vulnerable to a message stream modification attack; in other words, it is possible for an opponent, by making the necessary changes to the checksum portion of the message as well as to the data portion, to alter the ciphertext in such a way that it is not detected by an error-correcting code.

OUTPUT FEEDBACK MODE (OFB)



Output Feedback (OFB) mode encryption



Output Feedback (OFB) mode decryption

OUTPUT FEEDBACK MODE (OFB)

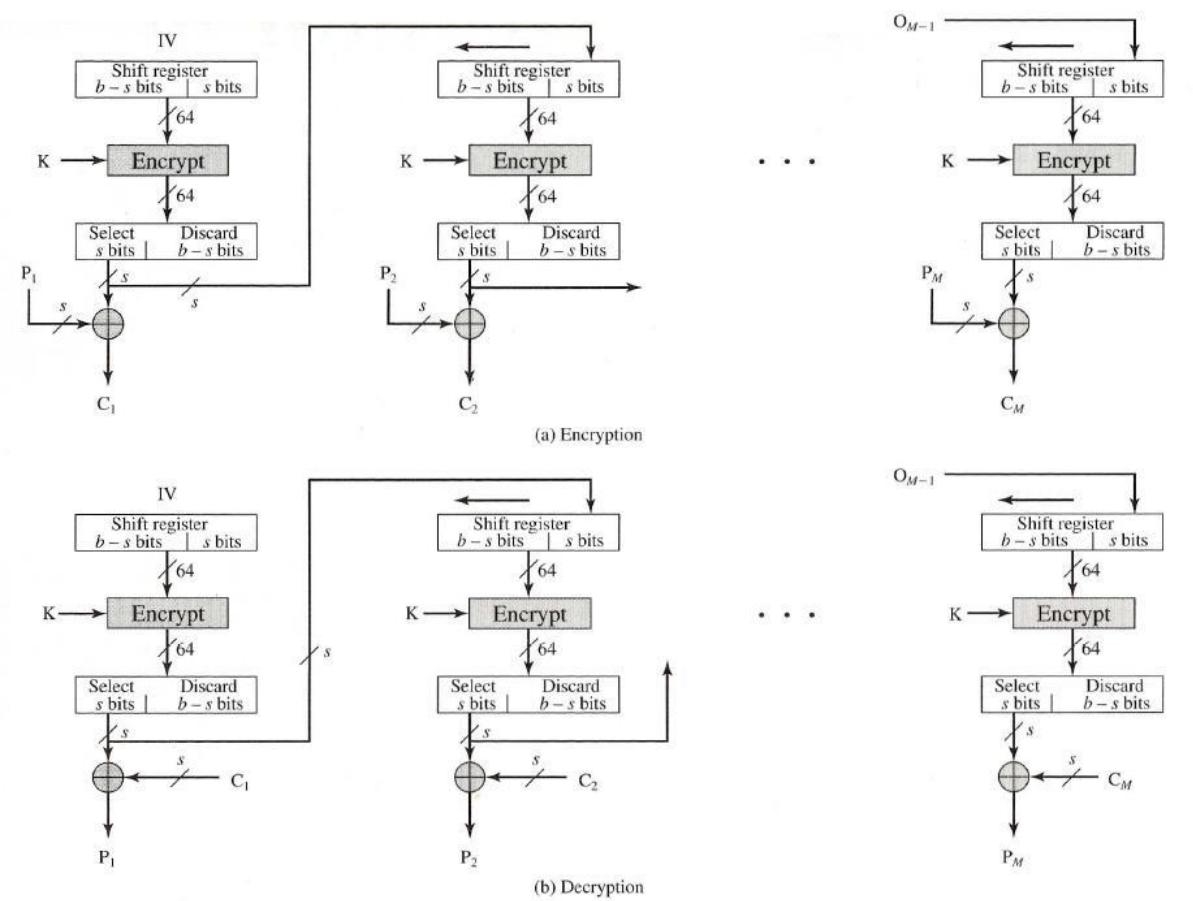


Figure 6.6 *s*-bit Output Feedback (OFB) Mode

OUTPUT FEEDBACK MODE (OFB)

- The encryption and decryption formula are as follows:

$$C_i = P_i \oplus S_s[E(K, O_{i-1})]$$

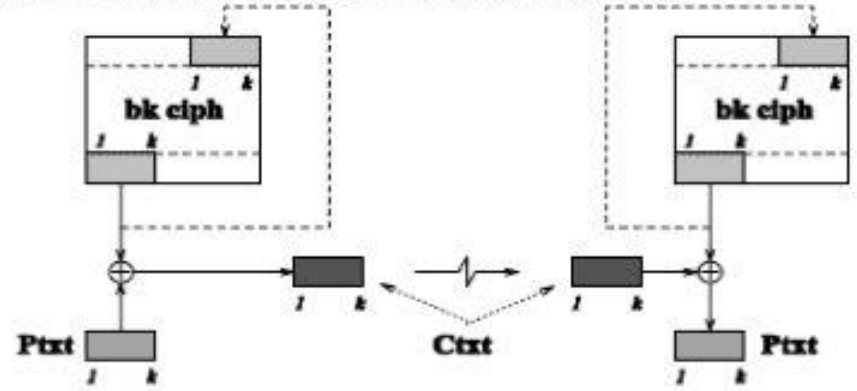
$$P_i = C_i \oplus S_s[E(K, O_{i-1})]$$

$$C_0 = IV$$

OUTPUT FEEDBACK MODE (OFB)

Input	000	001	010	011	100	101	110	111
Output	111	110	011	100	001	000	101	010

2-bit OFB
IV=111

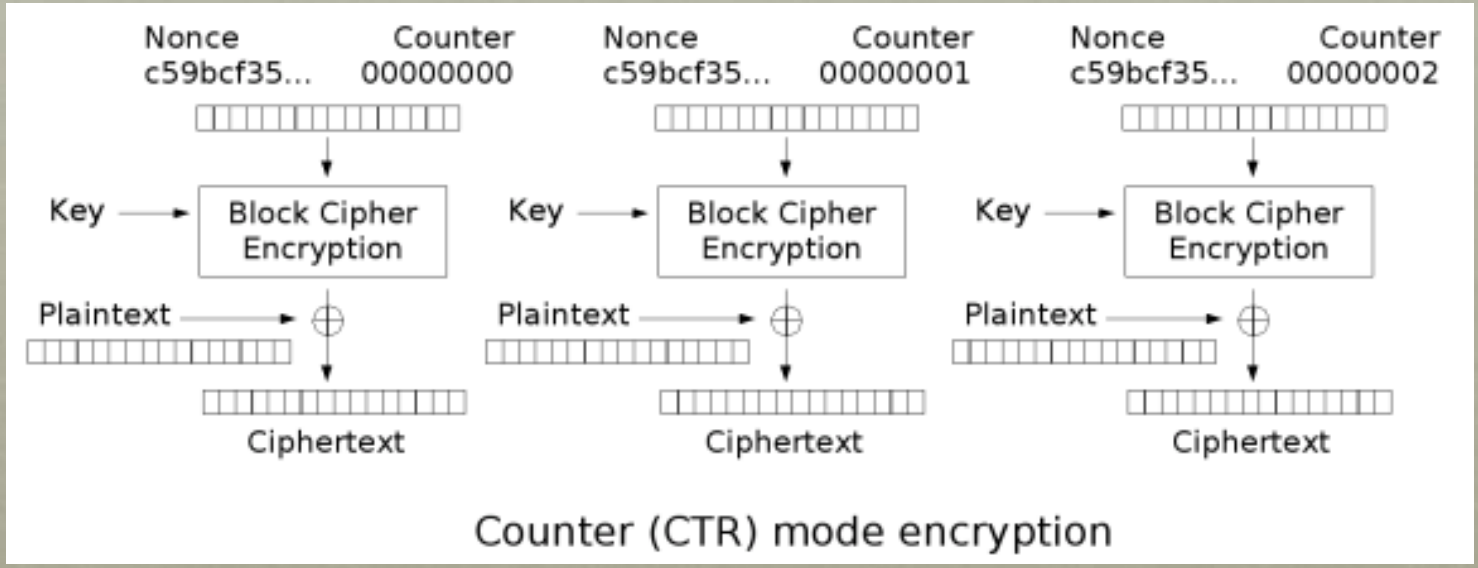


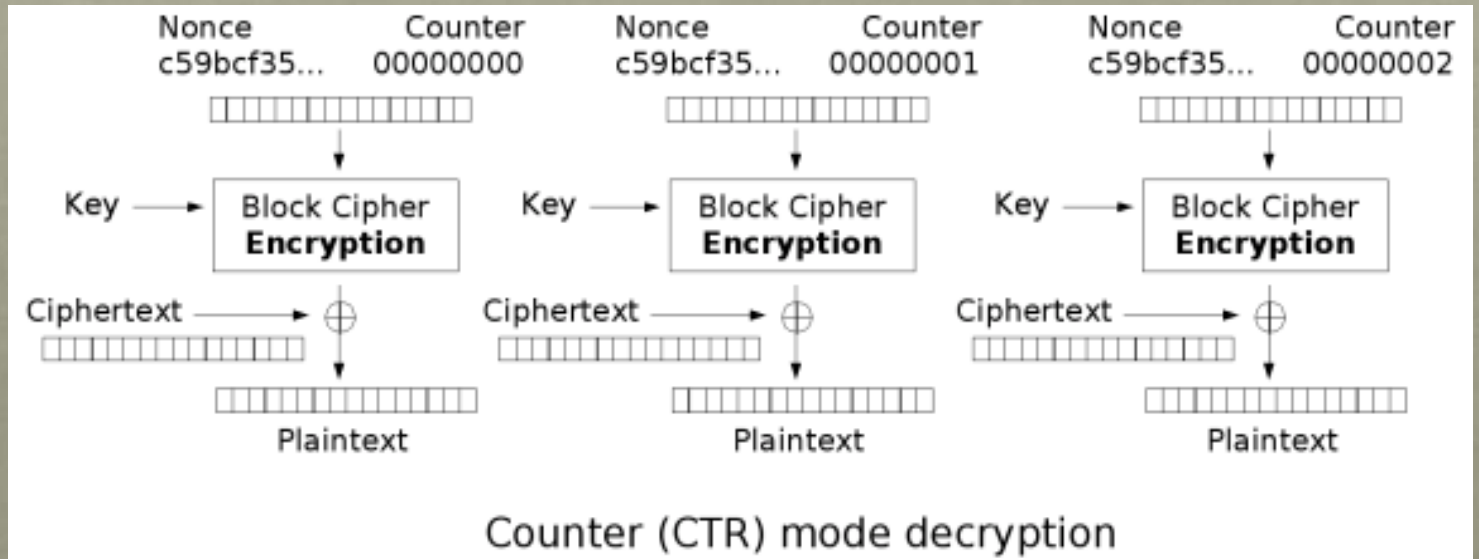
Plaintext	10	11	01	11	00	10
Input	111	**101**	**100**	**000**	**011**	**110**
E(Input)	**010**	**000**	**001**	**111**	**100**	**101**
Ciphertext	11	11	01	00	10	00

COUNTER MODE (CTR)

- Counter mode (CTR) is also known as Integer Counter Mode (ICM) and Segmented Integer Counter Mode (SIC).
- In this mode, a counter, equal to the plaintext block size is used to encrypt the plaintext blocks.
- 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.
- Parallel encryption is possible

COUNTER MODE (CTR)





COUNTER MODE (CTR)

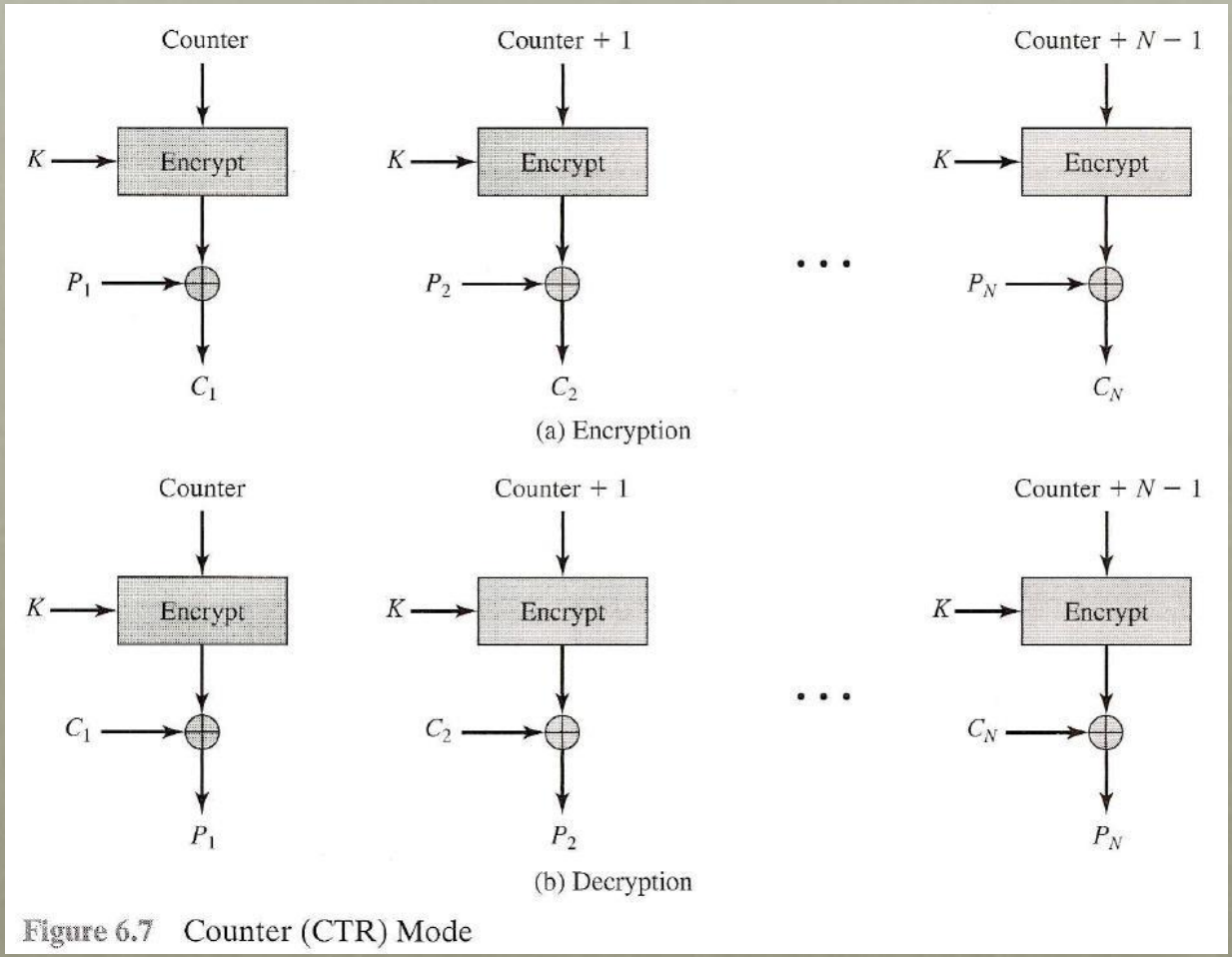


Figure 6.7 Counter (CTR) Mode

COUNTER MODE (CTR)

- To encrypt, the counter is encrypted and then XORed with the plaintext block to produce the ciphertext block; there is no chaining.
- To decrypt, the same sequence of counter values is used, with each encrypted counter XORed with a ciphertext block to recover the corresponding plaintext block.

COUNTER MODE (CTR)

- Advantages:
  - Hardware efficiency
    - Encryption can be done in parallel on multiple blocks of plaintext or ciphertext. Thus the throughput is only limited by the amount of parallelism that is achieved.
  - Software efficiency
    - Algorithms such as aggressive pipelining, multiple instruction dispatch per clock cycle, and SIMD instructions can be effectively utilized.

COUNTER MODE (CTR)

- Preprocessing
  - The execution of the underlying encryption algorithm does not depend on input of the plaintext or ciphertext, thus, preprocessing can be used to prepare the output of the encryption boxes that feed into the XOR functions.
- Random access
  - The i^{th} block of plaintext or ciphertext can be processed in random-access fashion. This benefits applications in which a ciphertext is stored and it is desired to decrypt just one block.

COUNTER MODE (CTR)

- Simplicity
  - Like CFB and OFB, only the encryption algorithm (function) is required.

COUNTER MODE (CTR)

Input	000	001	010	011	100	101	110	111
Output	111	110	011	100	001	000	101	010

CTR

IV=000

$$\text{Counter}_i = \text{Counter}_{i-1} + 1$$

Plaintext	101	101	110	010
IV	000	001	010	011
E(IV)	111	110	011	100
Ciphertext	010	011	101	110

Block Cipher Modes of Operation (Summary)

Mode	Description	Typical Application
Electronic Codebook (ECB)	Each block of 64 plaintext bits is encoded independently using the same key.	• 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 64 bits of plaintext and the preceding 64 bits of ciphertext.	• General-purpose block-oriented transmission. • Authentication

Block Cipher Modes of Operation (Summary)

Mode	Description	Typical Application
Cipher Feedback (CFB)	Input is processed *j* 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.	• General-purpose stream-oriented transmission. • Authentication.

Block Cipher Modes of Operation (Summary)

Mode	Description	Typical Application
Output Feedback (OFB)	Similar to CPF, except that the input to the encryption algorithm is the preceding DES output.	• Stream-oriented transmission over noisy channel (e.g., satellite communication)
Counter (CTR)	Each block of plaintext is XORed with an encrypted counter. The counter is incremented for each subsequent block.	• General-purpose block-oriented transmission. • Useful for high-speed requirements.

STREAM CIPHER

- Uses internal state and that the i^{th} ciphertext unit depends on the i^{th} plaintext unit, the secret key, and some state.
- Two major classes of stream ciphers:
  - Synchronous stream cipher
  - Nonsynchronous stream cipher

STREAM CIPHER

- Operating a block cipher in OFB mode yields a synchronous (additive) stream cipher; that is the next state does not depend on previously generated ciphertext units.
- Operating a block cipher in CFB mode yields a nonsynchronous (self-synchronizing) stream cipher; that is the next state depends on previously generated ciphtertext units.