National Chung Cheng University

Department of Information Management

Block Cipher Operation

Pei-Ju (Julian) Lee

National Chung Cheng University

Information Security

pjlee@mis.ccu.edu.tw

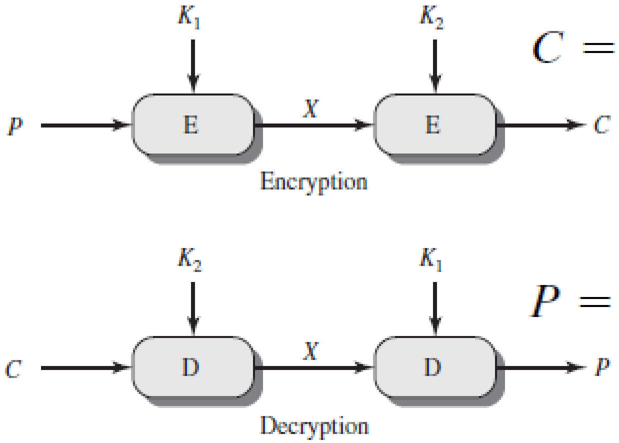
Fall, 2016

Multiple Encryption and Triple DES

- The potential vulnerability of DES to a brute-force attack
- Alternatives:
  - A completely new algorithm, Ex. AES
  - Preserve the existing investment in software and equipment, EX. use multiple encryption with DES and multiple keys

Double DES

- Two encryption stages and two keys K_1 and K_2
- Key length of 56*2 =112 bits, increase in cryptographic strength



(a) Double encryption

$$C = \mathrm{E}(K_2, \mathrm{E}(K_1, P))$$

$$P = \mathrm{D}(K_1, \mathrm{D}(K_2, C))$$

$$\mathrm{E}(K_2, \mathrm{E}(K_1, P)) = \mathrm{E}(K_3, P)$$

?

Cont'd

- Consider that encryption with DES is a mapping of 64-bit blocks to 64-bit blocks. (Mapping => permutation)
  - With 2^{64} possible inputs, will have $(2^{64})!$ different permutation
  - Or, in addition, one mapping for each different key (key length: 56 bits), for a total number of mappings : 2^{56}
- Therefore, Double DES with different keys will produce one of the many mappings that are not defined by a single application of DES

Meet-in-the-Middle Attack

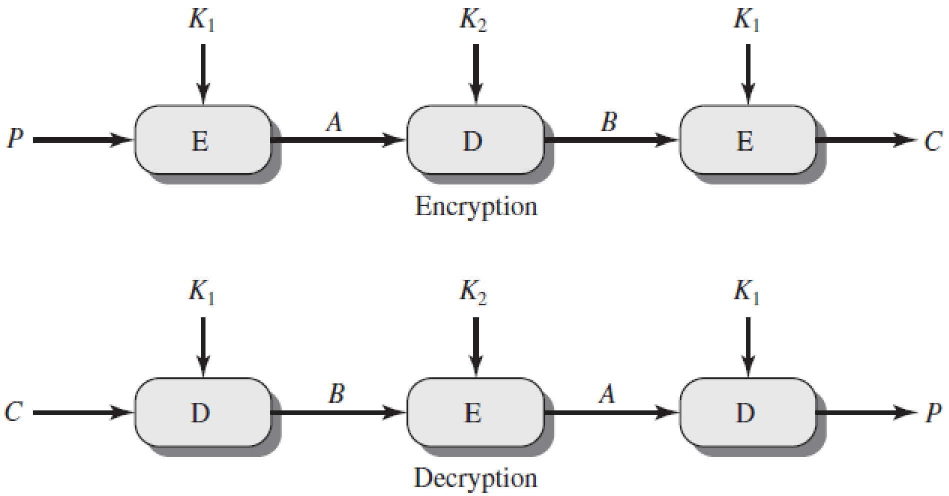
$$C = \mathrm{E}(K_2, \mathrm{E}(K_1, P)) \qquad X = \mathrm{E}(K_1, P) = \mathrm{D}(K_2, C)$$

- Given a known pair, (*P*, *C*)
  - Encrypt for all 2^{56} possible values of K_1 $\boxed{E(K_1, P)}$
  - Store these results in a table and then sort the table by the values of X
  - Decrypt *C* using all 2^{56} possible values of K_2 $\boxed{D(K_2, C)}$
  - Check the result against the table for a match
  - If a match occurs, then test the two resulting keys against a new known plaintext–ciphertext pair. If the two keys produce the correct *C*, accept them as the correct keys

Cont'd

- This attack will work against any block encryption cipher
- For any given P (64 bits input) of Double DES:
  - there are 2^{64} possible C values
  - 2^{112} possible keys for double keys (each 56 bits)
  - Therefore, on average, the number of different 112-bit keys that will produce a given ciphertext C is $2^{112}/2^{64} = 2^{48}$
  - Thus, the procedure will produce about 2^{48} false alarms on the first (P, C) pair. But with an additional 64 bits of known plaintext and ciphertext, the false alarm rate is reduced to $2^{48-64} = 2^{-16}$
  - The probability that the correct keys are determined is $1 - 2^{-16}$. The result is that a known plaintext attack will succeed against double DES, which has a key size of 112 bits, with an effort on the order of 2^{56}, which is not much more than the 2^{55} required for single DES

Triple-DES with Two-Keys



Encryption

Decryption

(b) Triple encryption

$$C = \mathrm{E}(K_1, \mathrm{D}(K_2, \mathrm{E}(K_1, P)))$$
$$P = \mathrm{D}(K_1, \mathrm{E}(K_2, \mathrm{D}(K_1, C)))$$

Alternative to DES

3DES with two keys has been adopted for use in the key management standards ANSI X9.17 and ISO 8732.1

- Three stages of encryption with 3 different keys
  - Advantage: Raises the cost of the meet-in-the-middle attack to 2^{112} (Double DES is 2^{56})
  - Disadvantage: key length of 56 x 3 = 168 bits
  - Alternative: A triple encryption method that uses only two keys
  - The function follows an encrypt-decrypt-encrypt (EDE) sequence

Cont’d

- Decryption for the second stage in EDE:
  - no cryptographic significance
  - it allows users of 3DES to decrypt data encrypted by users of the older single DES:
  - $C = \mathrm{E}(K_1, \mathrm{D}(K_1, \mathrm{E}(K_1, P))) = \mathrm{E}(K_1, P)$
  - $P = \mathrm{D}(K_1, \mathrm{E}(K_1, \mathrm{D}(K_1, C))) = \mathrm{D}(K_1, C)$

Attacks on Two-keys 3DES

- Currently, there are no practical cryptanalytic attacks on Two-keys 3DES.
- The cost of a brute-force key search on Two-keys 3DES is on the order of 2^{112}
- Some attacks on Two-keys 3DES: [MERK81], known-plaintext [VANO90]
- Three-keys 3DES is preferred for higher security: key length of 168 bits

Triple DES with Three Keys

- Many researchers now feel that three-key 3DES is the preferred alternative

Three-key 3DES has an effective key length of 168 bits and is defined as:

- $C = E(K_3, D(K_2, E(K_1, P)))$

Backward compatibility with DES is provided by putting:

- $K_3 = K_2$ or $K_1 = K_2$

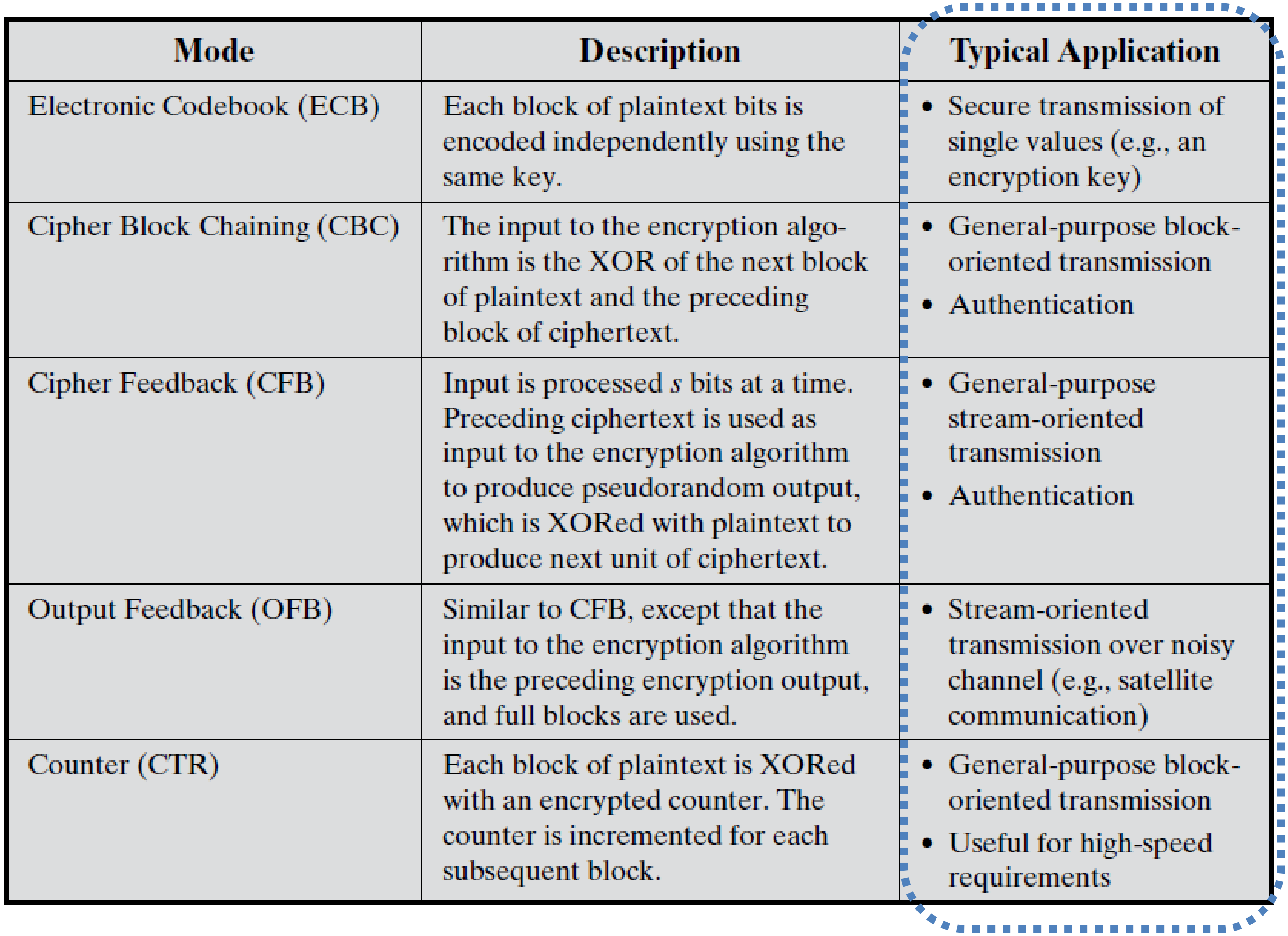
- A number of Internet-based applications have adopted three-key 3DES including PGP and S/MIME

Modes of Operation

- To apply a block cipher in a variety of applications, five modes of operation have been defined by NIST
  - In essence, a mode of operation is a technique for enhancing the effect of a cryptographic algorithm or adapting the algorithm for an application, such as applying a block cipher to a sequence of data blocks or a data stream.
  - The five modes are intended to cover a wide variety of applications of encryption for which a block cipher could be used

Block Cipher Modes of Operation

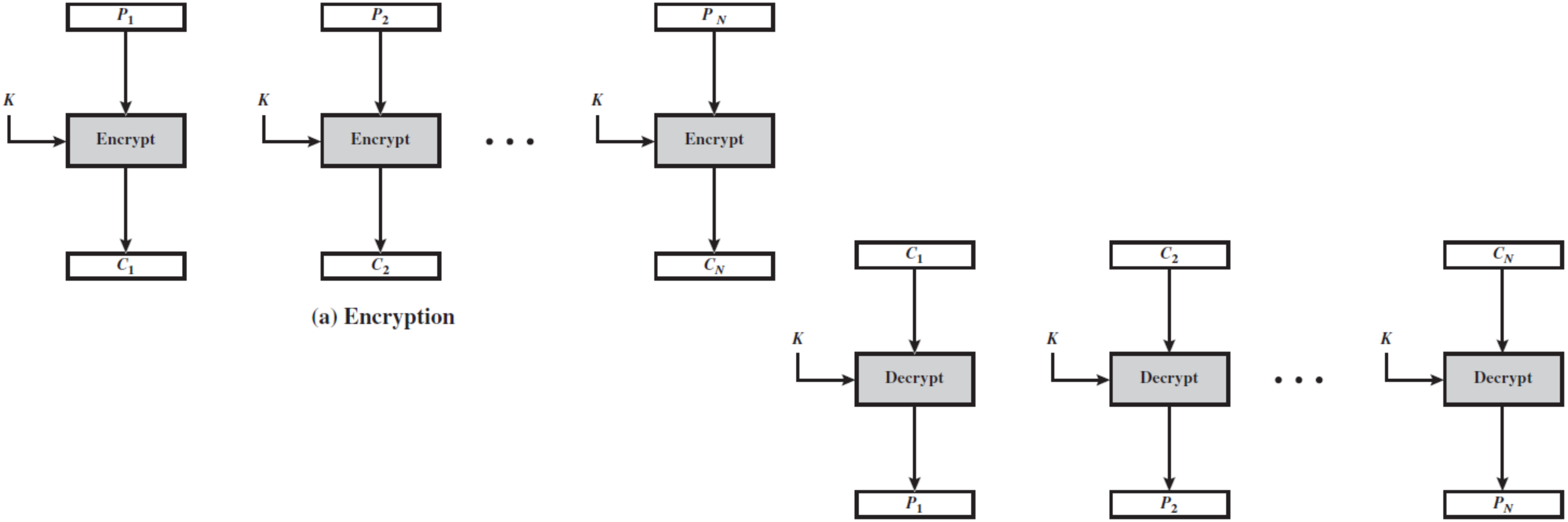
- A mode of operation is a technique for enhancing the effect of a cryptographic algorithm
- These modes are intended for use with any symmetric block cipher, including triple DES and AES



Mode	Description	Typical Application
Electronic Codebook (ECB)	Each block of 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 block of plaintext and the preceding block of ciphertext.	• General-purpose block-oriented transmission • Authentication
Cipher Feedback (CFB)	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.	• General-purpose stream-oriented transmission • Authentication
Output Feedback (OFB)	Similar to CFB, except that the input to the encryption algorithm is the preceding encryption output, and full blocks are used.	• 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

ECB

- Electronic codebook (ECB) mode
  - Plaintext is handled one block at a time and each block is encrypted using the same key
  - The term codebook: for a given key, there is a unique ciphertext for every b-bit block of plaintext

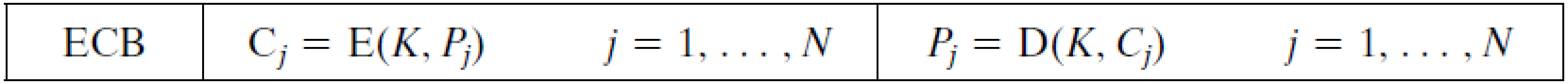


(a) Encryption

(b) Decryption

Cont’d

- For a message longer than *b* bits, 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.

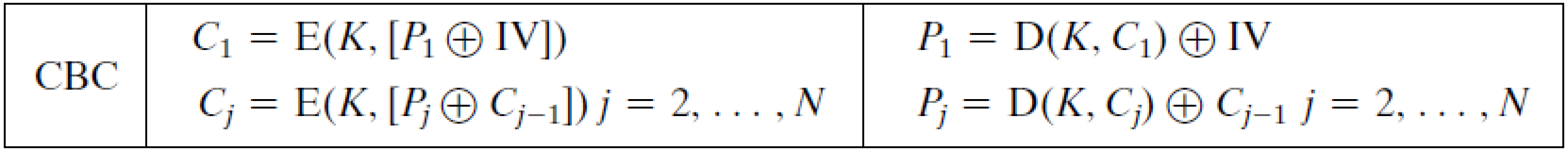


ECB	$C_j = E(K, P_j) \qquad j = 1, \ldots, N$	$P_j = D(K, C_j) \qquad j = 1, \ldots, N$

- Suitable for a short amount of data, such as an encryption key.
  - E.g. transmit a DES or AES key
- Disadvantage: the same *b*-bit block of plaintext produces the same ciphertext.

CBC

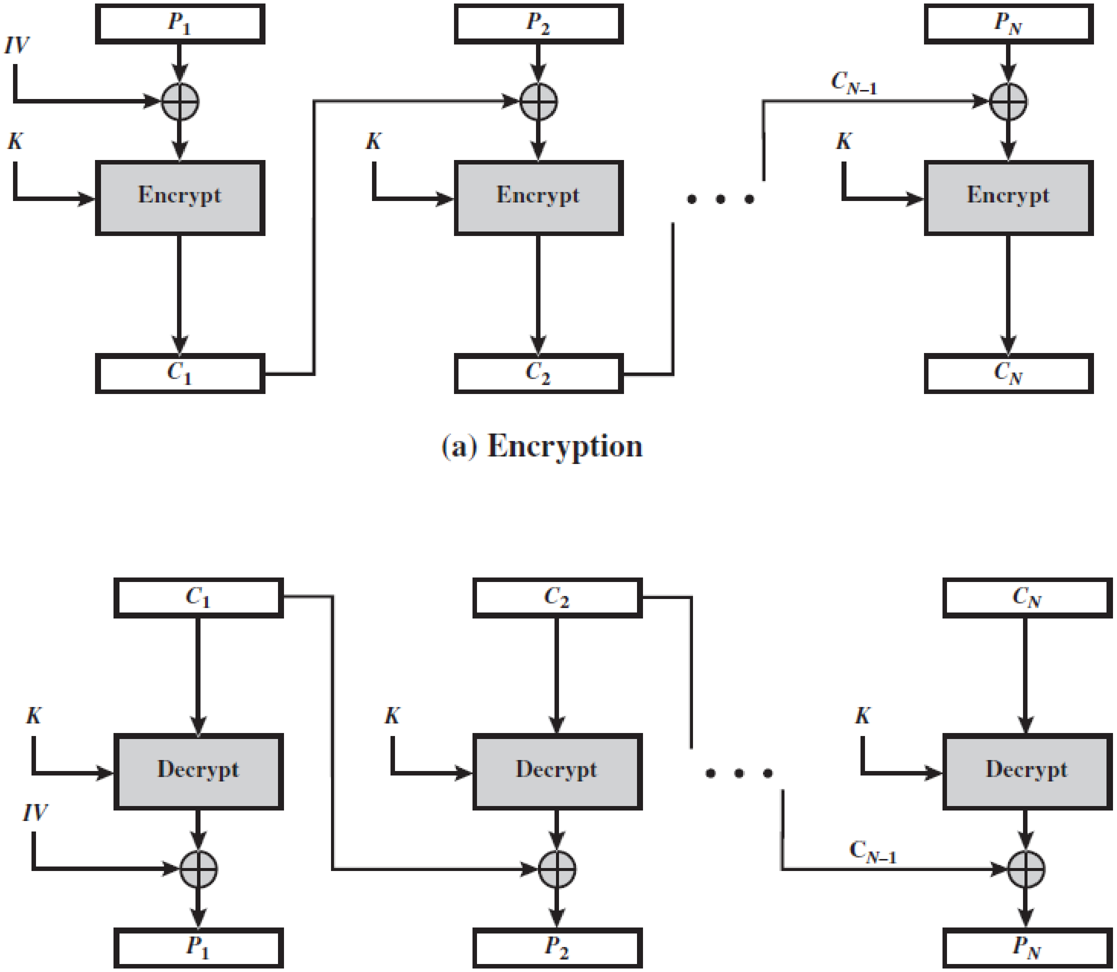
- Cipher block chaining (CBC) mode
  - The input to the encryption algorithm is the XOR of the current plaintext block and the preceding ciphertext block
  - The same key is used for each block
  - Therefore, repeating plaintext input has no relationship to the ciphertext



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

Cont’d

Initialization vector (IV): most be known to both the sender and receiver; but should be protected as protecting the key for maximum security

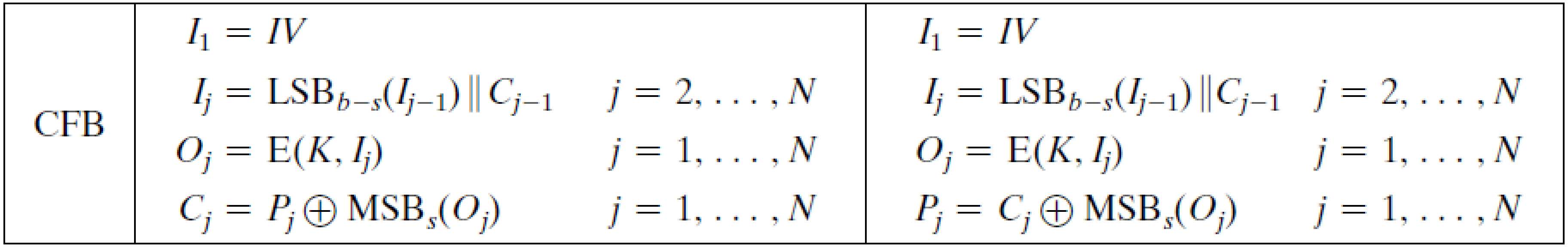


(a) Encryption

(b) Decryption

CFB

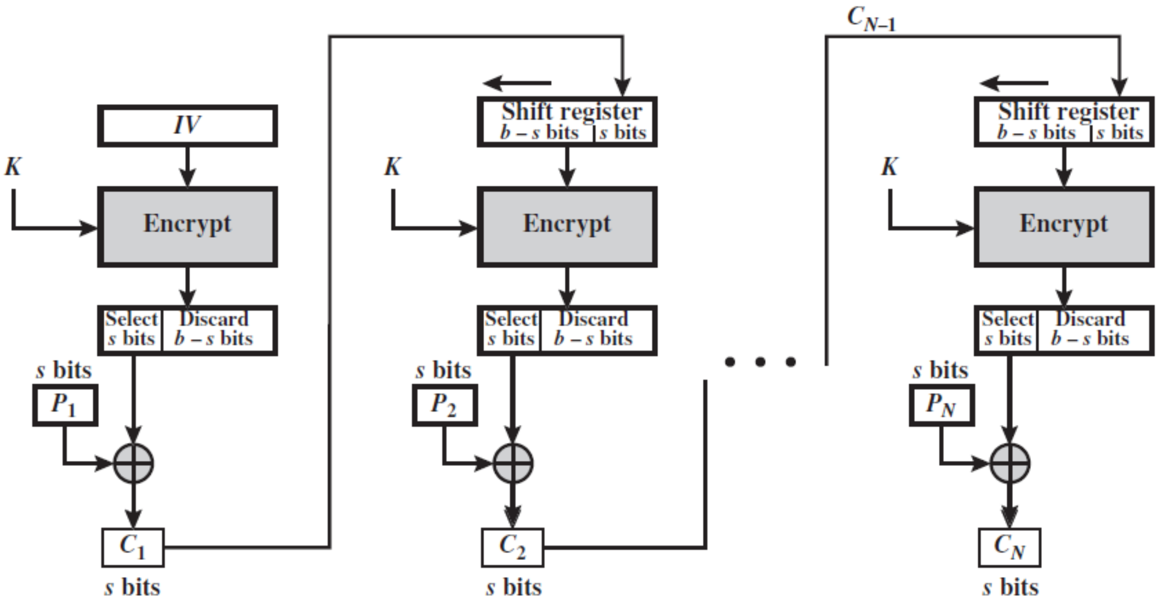
- Cipher feedback (CFB) mode
  - The plaintext is divided into segments of *s* bits for transmission(e.g. *s = 8*)



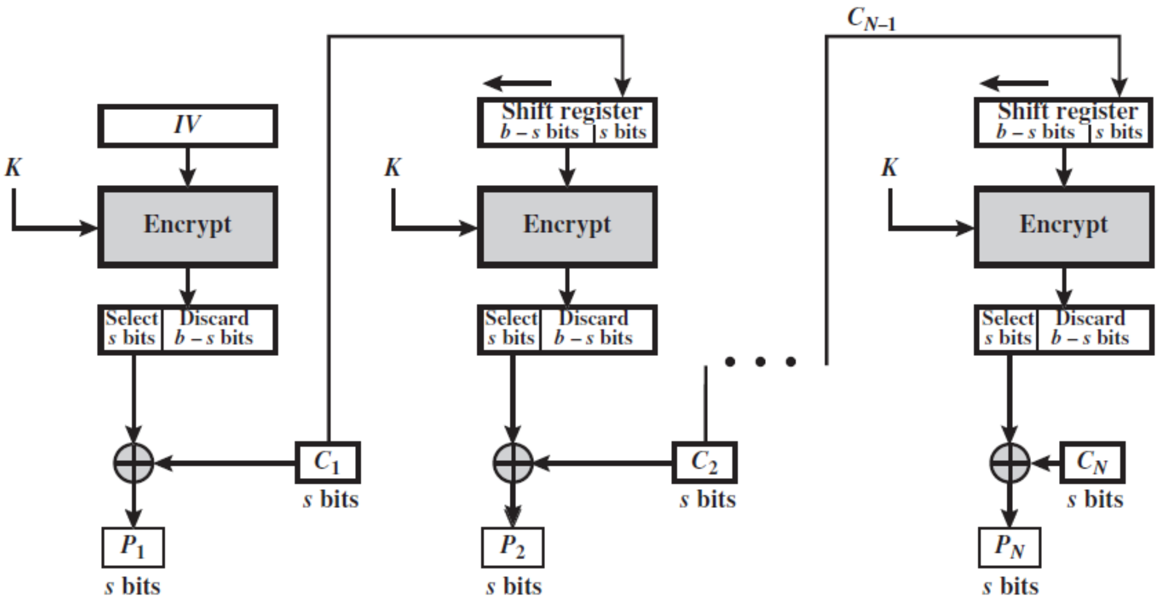
CFB	$I_1 = IV$		$I_1 = IV$	
	$I_j = \mathrm{LSB}_{b-s}(I_{j-1}) \| C_{j-1}$	$j = 2, \ldots, N$	$I_j = \mathrm{LSB}_{b-s}(I_{j-1}) \| C_{j-1}$	$j = 2, \ldots, N$
	$O_j = \mathrm{E}(K, I_j)$	$j = 1, \ldots, N$	$O_j = \mathrm{E}(K, I_j)$	$j = 1, \ldots, N$
	$C_j = P_j \oplus \mathrm{MSB}_s(O_j)$	$j = 1,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

  - Using the CFB mode, OFB mode, and CTR mode can convert a block cipher into a stream cipher
    - Eliminates the need to pad a message to be an integral number of blocks
    - Can operate in real time

- LSB(least significant bit)(rightmost bit)
- MST(most significant bit)(leftmost bit)



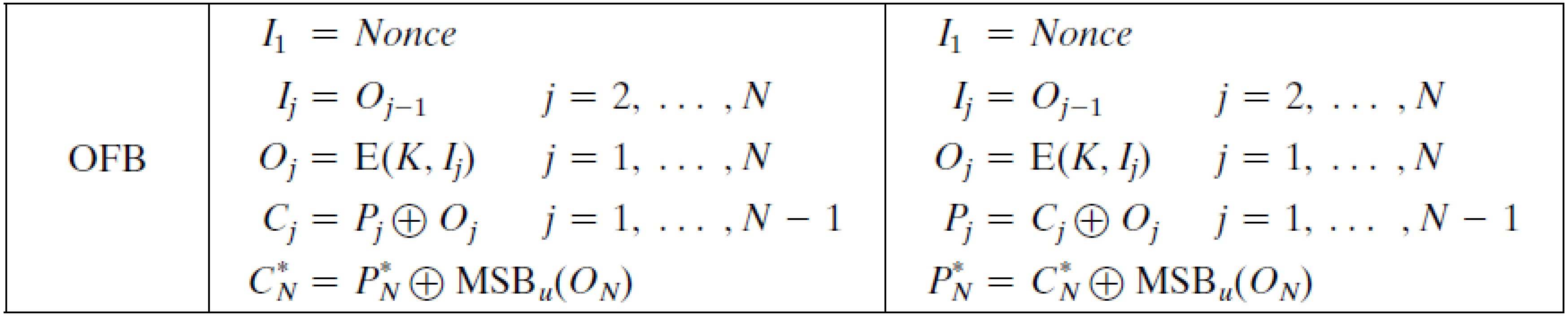
(a) Encryption



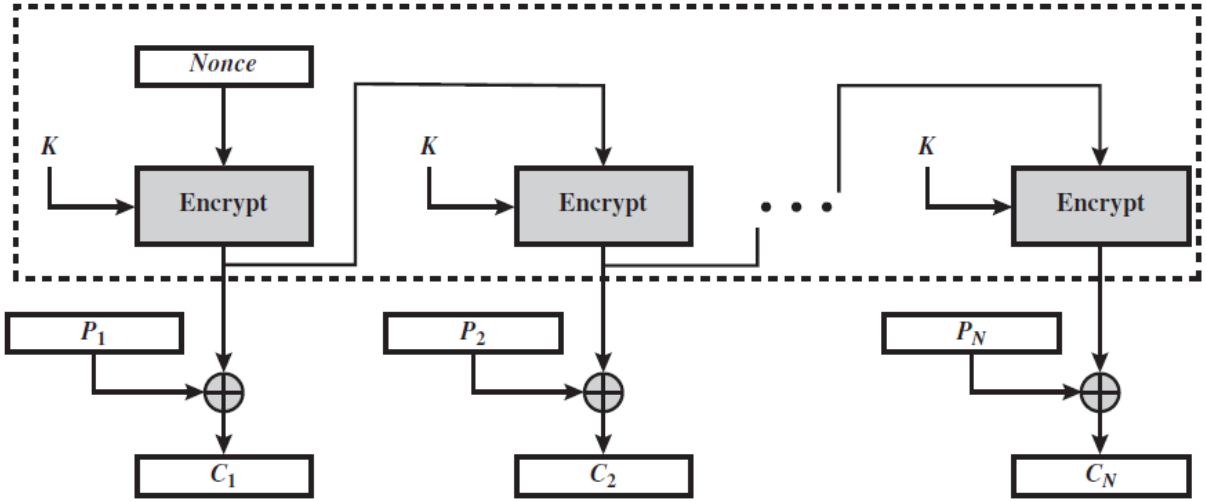
(b) Decryption

OFB

- Output feedback (OFB) mode
  - similar in structure to that of CFB
    - For OFB, the output of the encryption function is fed back to become the input for encrypting the next block of plaintext
    - In CFB, the output of the XOR unit is fed back to become input for encrypting the next block
    - OFB mode operates on full blocks of P and C, whereas CFB operates on an **s**-bit subset.

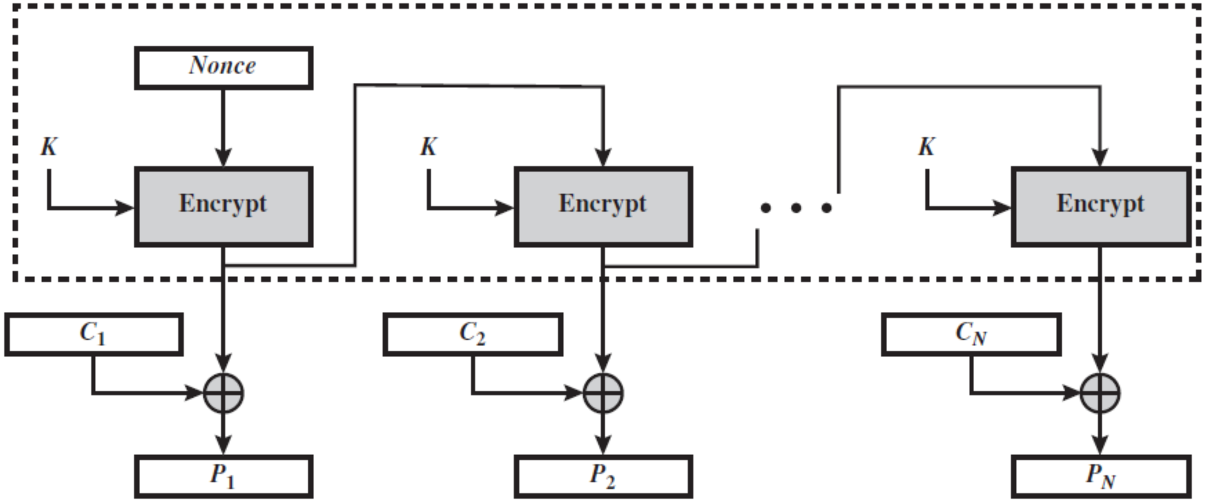


OFB	$I_1 = Nonce$ $I_j = O_{j-1} \quad j = 2, \ldots, N$ $O_j = \mathrm{E}(K, I_j) \quad j = 1, \ldots, N$ $C_j = P_j \oplus O_j \quad j = 1, \ldots, N - 1$ $C_N^* = P_N^* \oplus \mathrm{MSB}_u(O_N)$	$I_1 = Nonce$ $I_j = O_{j-1} \quad j = 2, \ldots, N$ $O_j = \mathrm{E}(K, I_j) \quad j = 1, \ldots, N$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



Generated stream part

(a) Encryption



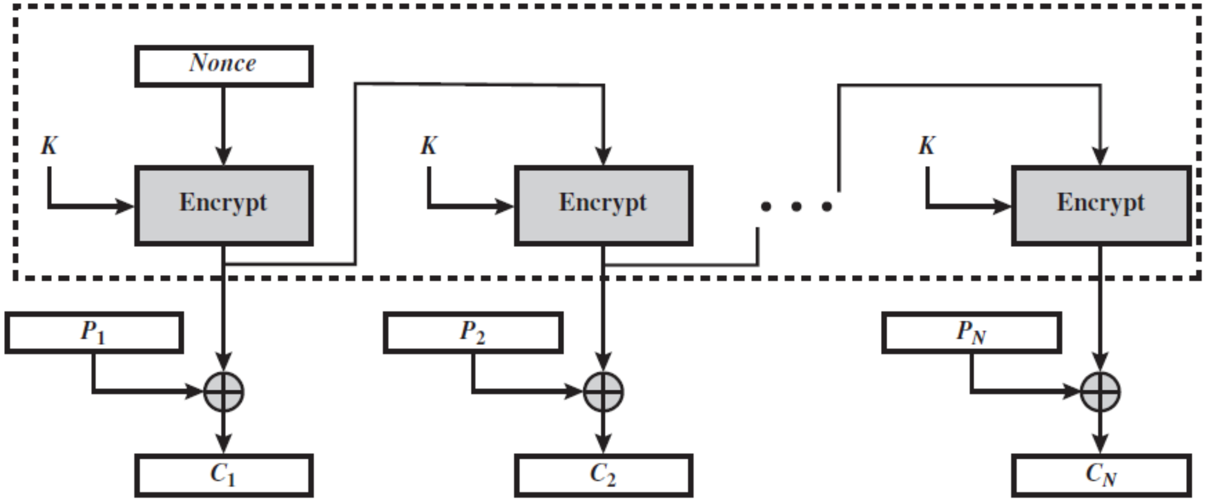
(b) Decryption

Cont'd

- Advantage of the OFB: bit errors in transmission do not propagate
  - E.g. if a bit error occurs in C_1, only the recovered value of P_1 is affected
- Disadvantage of OFB: more vulnerable to a message stream modification attack than is CFB
  - E.g. Consider that complementing a bit in the C complements the corresponding bit in the recovered P. Thus, controlled changes to the recovered P can be made. This may make it 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 C in such a way that it is not detected by an error-correcting code.

Cont’d

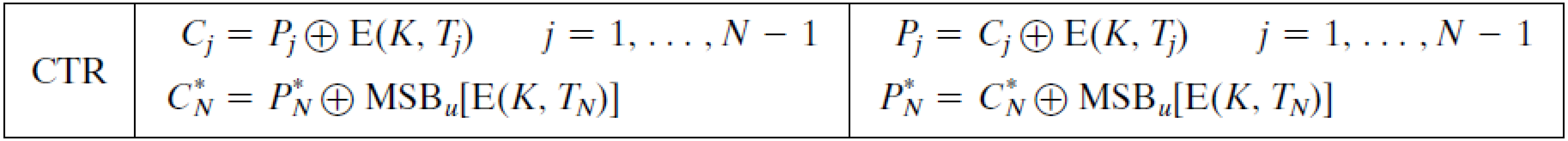
- OFB has the structure of a typical stream cipher, because the cipher generates a stream of bits as a function of an initial value and a key, and that stream of bits is XORed with the plaintext bits
- The generated stream that is XORed with the plaintext is itself independent of the plaintext



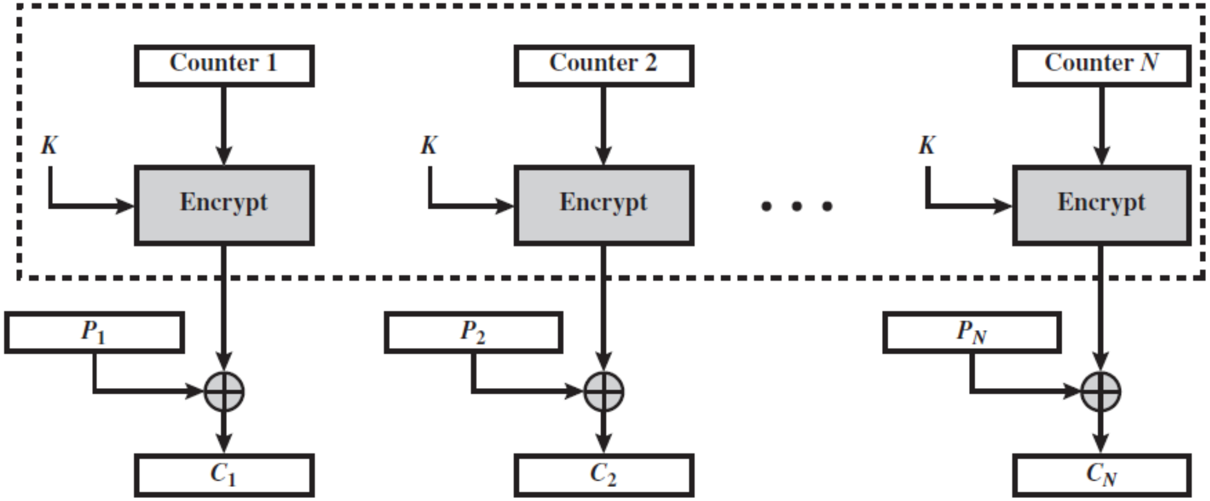
(a) Encryption

CTR

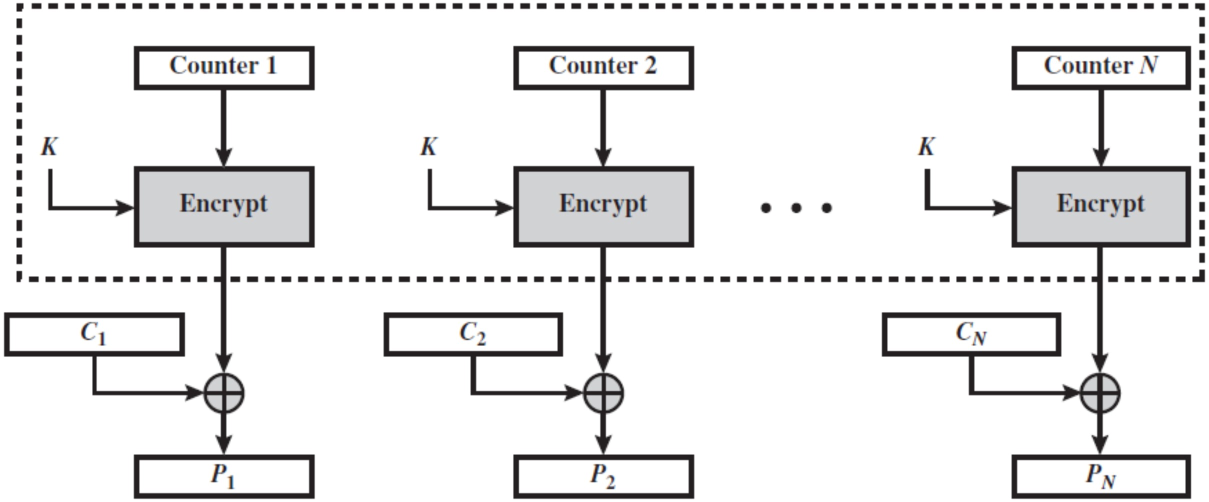
- Counter (CTR) mode
  - A counter equal to the plaintext block size is used
  - The counter value must be different for each plaintext block that is encrypted (incremented by 1 for each subsequent block)
  - For encryption, the counter is encrypted and then XORed with the plaintext block to produce the ciphertext block; there is no chaining
  - For decryption, the initial counter value must be made available



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



(a) Encryption



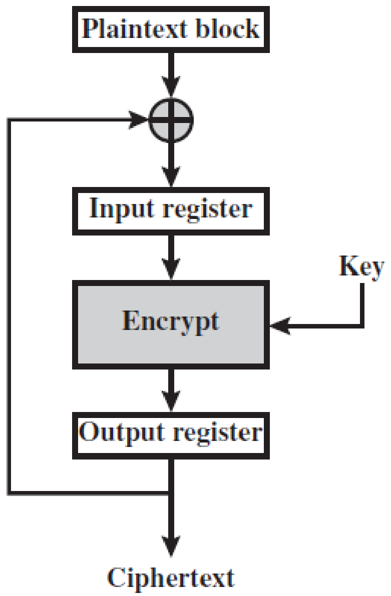
(b) Decryption

Cont’d

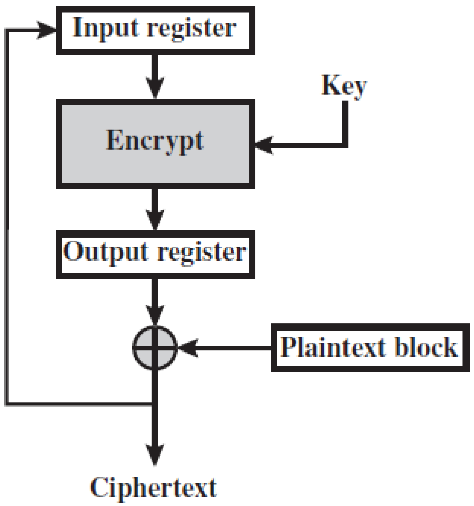
- Unlike the ECB, CBC, and CFB modes, we do not need to use padding in CTR
- If any plaintext block that is encrypted using a given counter value is known, then the output of the encryption function can be determined easily from the associated ciphertext block
  - One way to ensure the uniqueness of counter values is to continue to increment the counter value by 1 across messages.

Advantages of CTR Mode

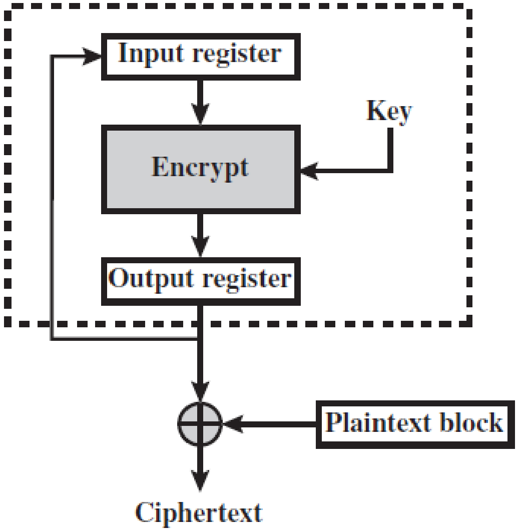
- Hardware/Software efficiency, Random access
  - Not in chaining mode, therefore, it can be done in parallel on multiple blocks of P/C
- Preprocessing:
  - The execution of the underlying encryption algorithm does not depend on input of the P/C
- Provable security
- Simplicity
  - CTR mode requires only the implementation of the encryption algorithm and not the decryption algorithm



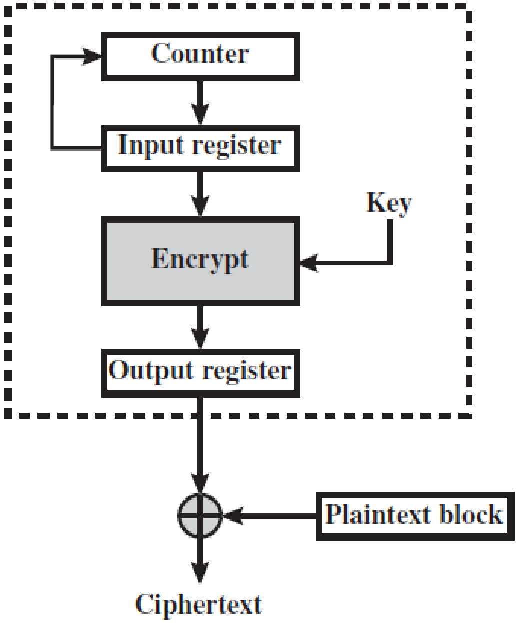
(a) Cipher block chaining (CBC) mode



(b) Cipher feedback (CFB) mode



(c) Output feedback (OFB) mode



(d) Counter (CTR) mode