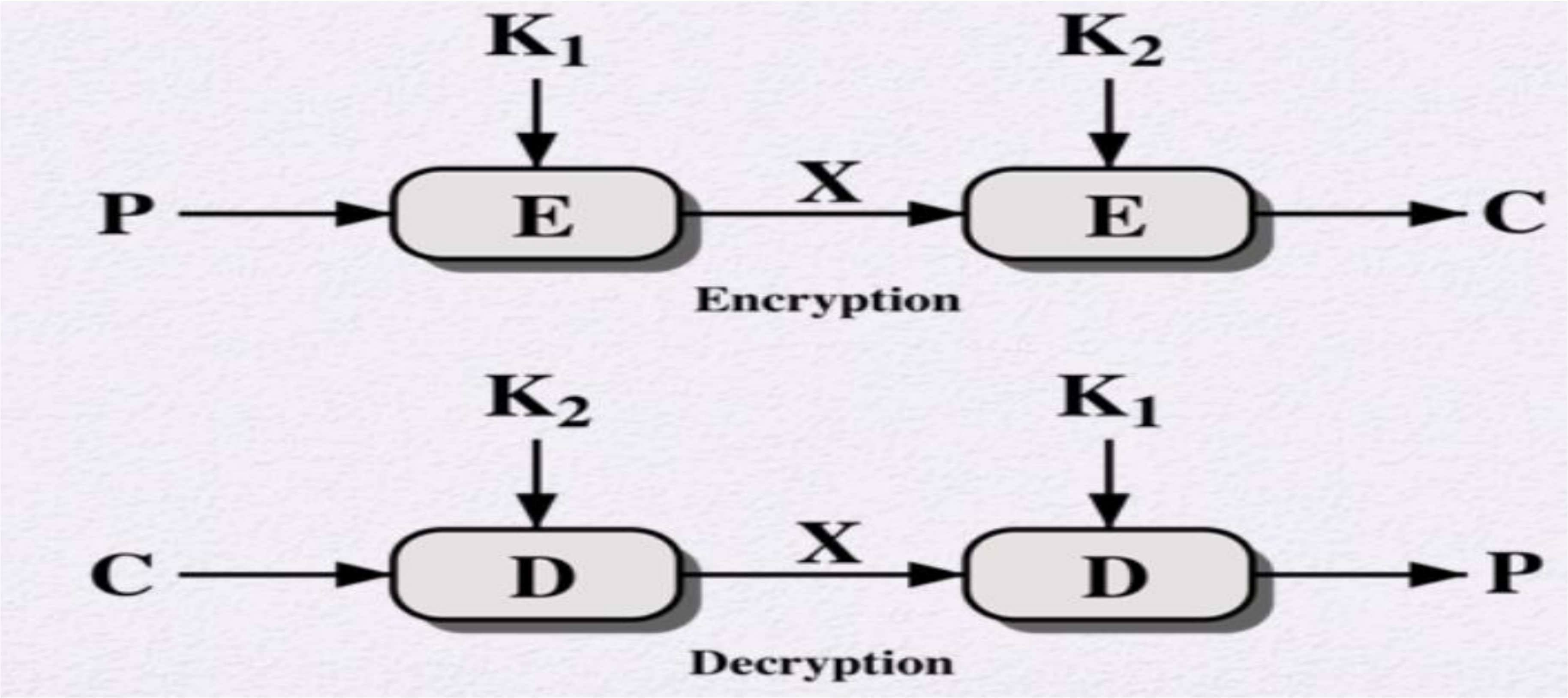
AES over DES

- AES was introduced as a replacement for DES.
- AES is more secure compared to DES.
- AES is more flexible compared to DES.
- The algorithmic structures of AES are simpler than that of DES.
- AES is more efficient compared to DES.
- AES suits for even resource-constrained devices.

MULTIPLE ENCRYPTIONS

Double DES

- $C = E(K_2, E(K_1, P))$
- $P = D(K_1, D(K_2, C))$
- Key Space = 2^{112}



Reduction of Double DES to a Single Stage

- Assume that $E(K_2,E(K_1, P)) = E(K_3, P)$

- However, each PT block is uniquely mapped to a CT block, and vice-versa.

- Moreover, the complex operations of DES will make the reduction to single stage almost impossible.

Meet in The Middle(MIM) Attack

- The attacker somehow gets to know (PT_i, CT_i)
- $E(K_1, P) = X = D(K_2, C)$
- Sort the encryption table by values of X.

K1	Output (X) = $EA(K_1, PTi)$
KE1	ACT1
KE2	ACT2
.	.
.	.
.	**X**
.	.
.	.
.	.
.	.
$KE_{2^{256}}$	$ACT_{2^{256}}$

K2	Output = $EA(K_1, PTi)$
KD1	ADT1
KD2	ADT2
.	.
.	.
.	.
.	.
.	.
.	.
.	**X**
$KD_{2^{256}}$	$ADT_{2^{256}}$

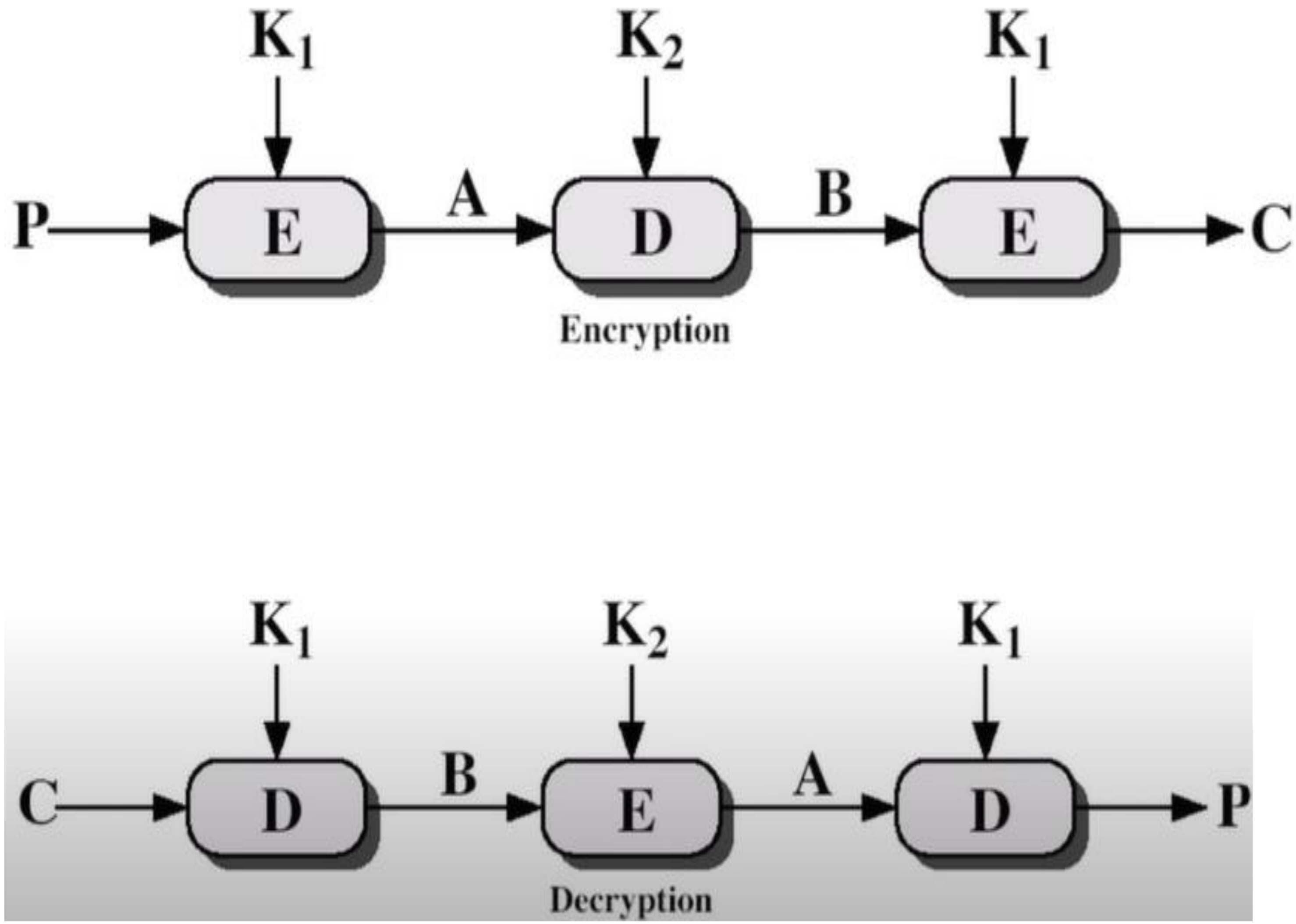
MIM on Double DES (Contd..)

- Practically, a Hash set search approach will be used to check if an element of second table exists in the first or not.
- Average Time complexity of Hash set search = O(1).
- Therefore, the strength of Double DES reduces to 2^{57} from the desired value 2^{112}.

Triple DES with 2 Keys

- Overcomes the disadvantage of Double DES.
- Double DES used the encryption functions in sequence.
- Triple DES follows E-D-E sequence.
- $C = E(K_1, D(K_2, E(K_1, P)))$
- $P = D(K_1, E(K_2, D(K_1, C)))$

Triple DES with 2 Keys



Known PT attack on Triple DES with 2 Keys

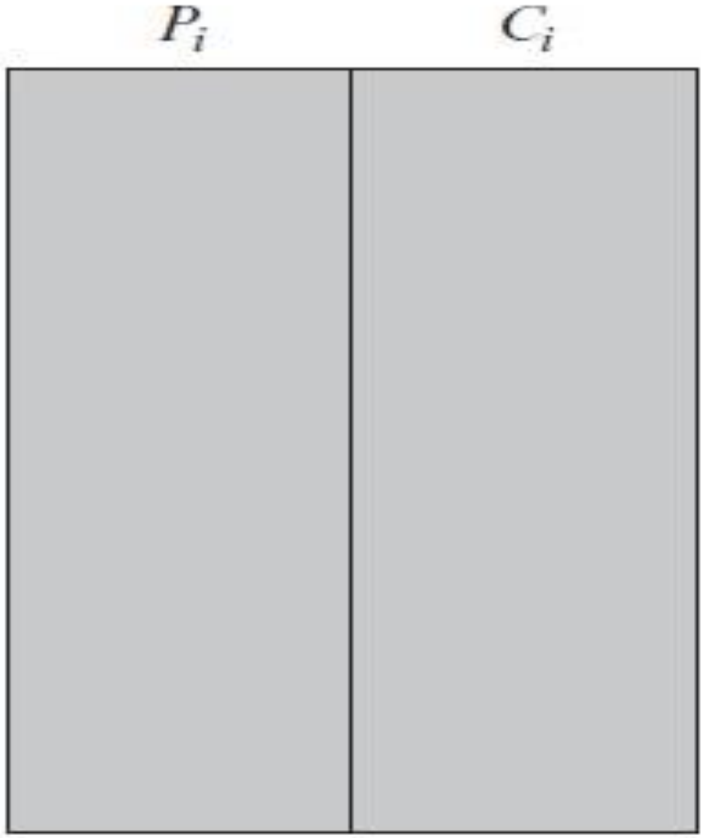


Table of n known plaintext–ciphertext pairs, sorted on P

- Pick any arbitrary value ‘a’ for A.
- For each $K_1 = i$, calculate $P_i = D(i, a)$.

Known PT attack on Triple DES with 2 Keys (Contd..)

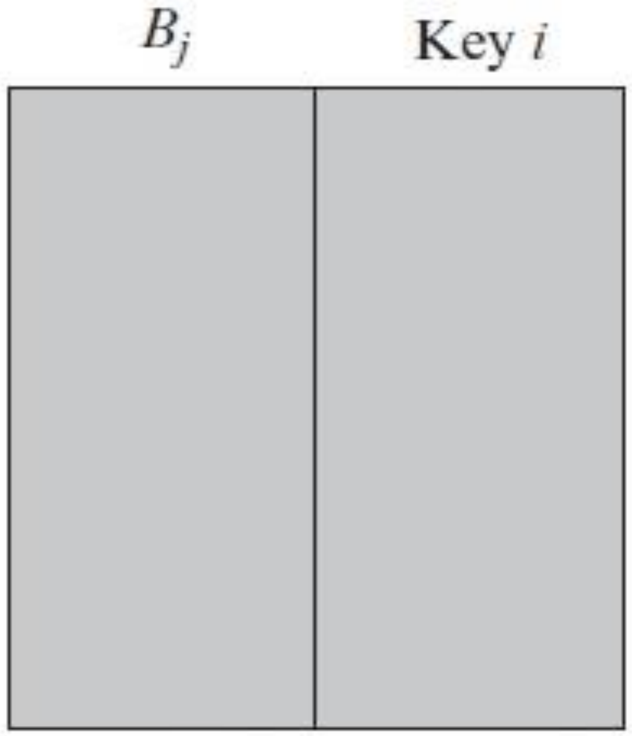
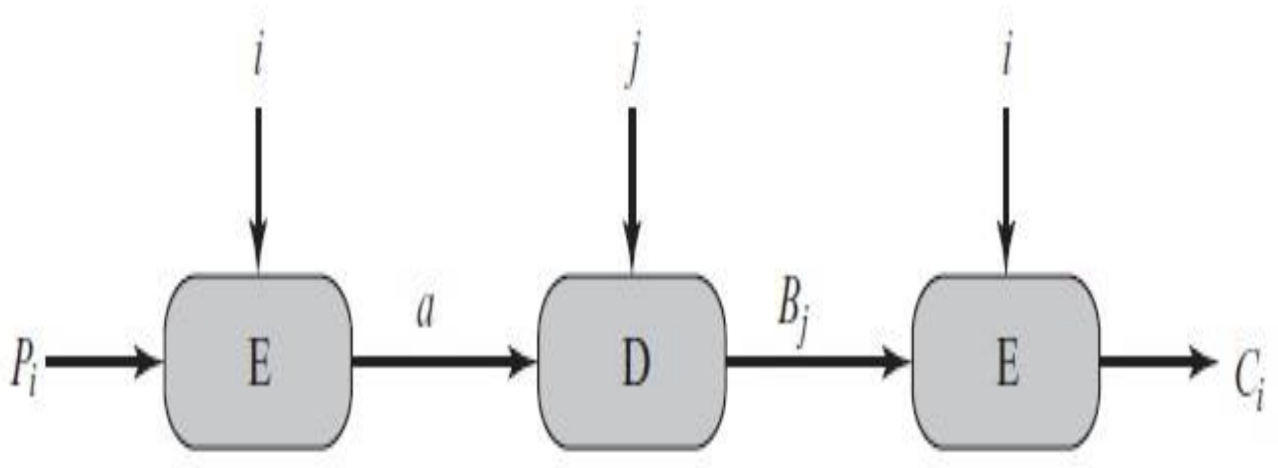


Table of intermediate values and candidate keys

- B = D(i, C)
- Sort the Table based on values of B.

Known PT attack on Triple DES with 2 Keys (Contd..)

- For each $K_2 = j$, calculate $B_j = D(j, a)$



Two-key triple encryption with candidate pair of keys

- Compute the pair (i, j) which produces the pair (P, C).
- Test each (i, j) to see if the desired CT is obtained or not for different (P, T) pairs. (Repeat the process with another ‘a’ if necessary).

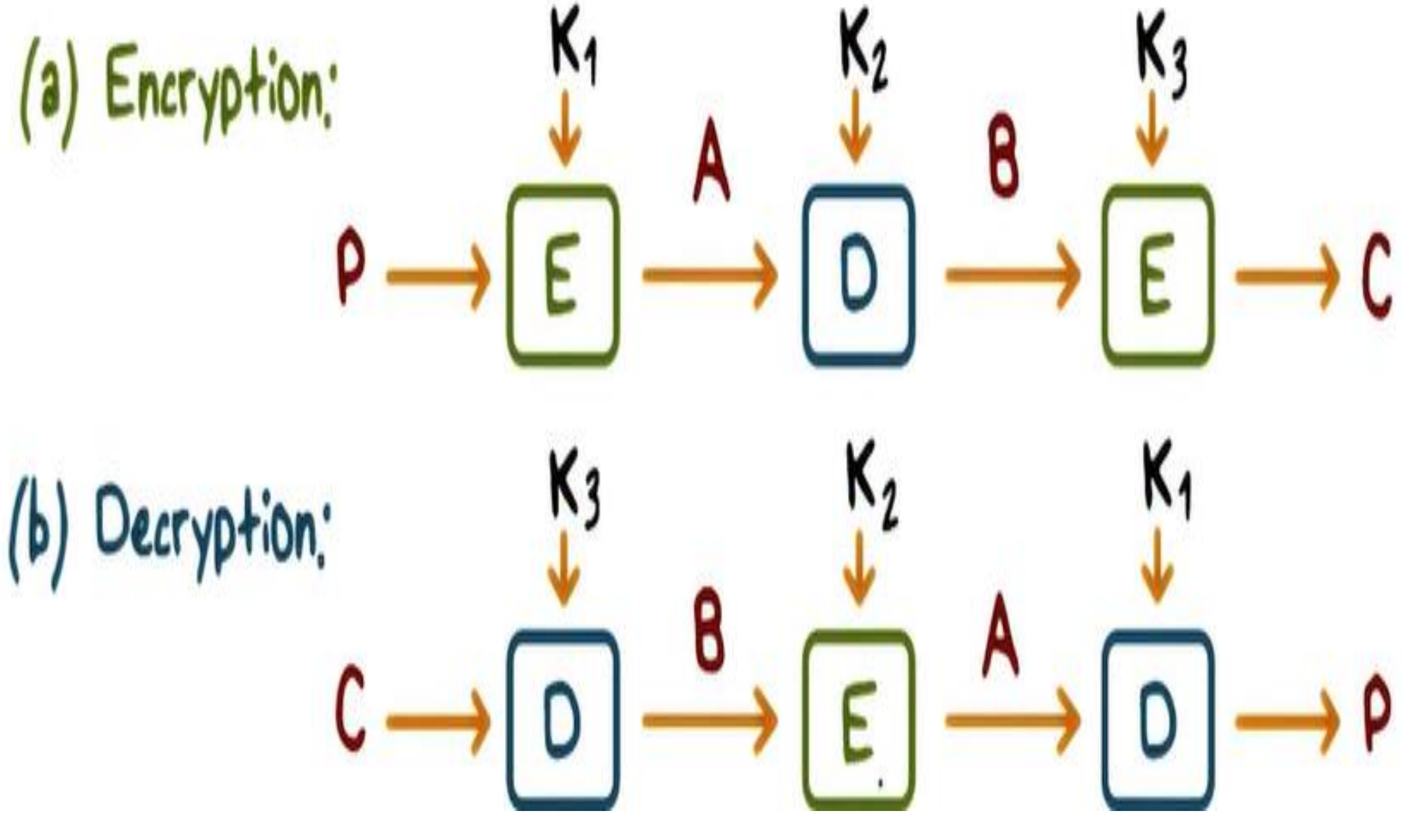
Feasibility of Known PT attack on Triple DES with 2 Keys

- For one (P, C) pair, the probability of success $= (1/2^{64})$.
- Hence, for 'n' (P, C) pairs, the probability of success $= n/2^{64}$.
- Expected number of values to be tried for 'a' for large 'n', with 1 key $= 2^{64}/n$.
- Expected complexity for running the attack $= 2^{120}/n$.

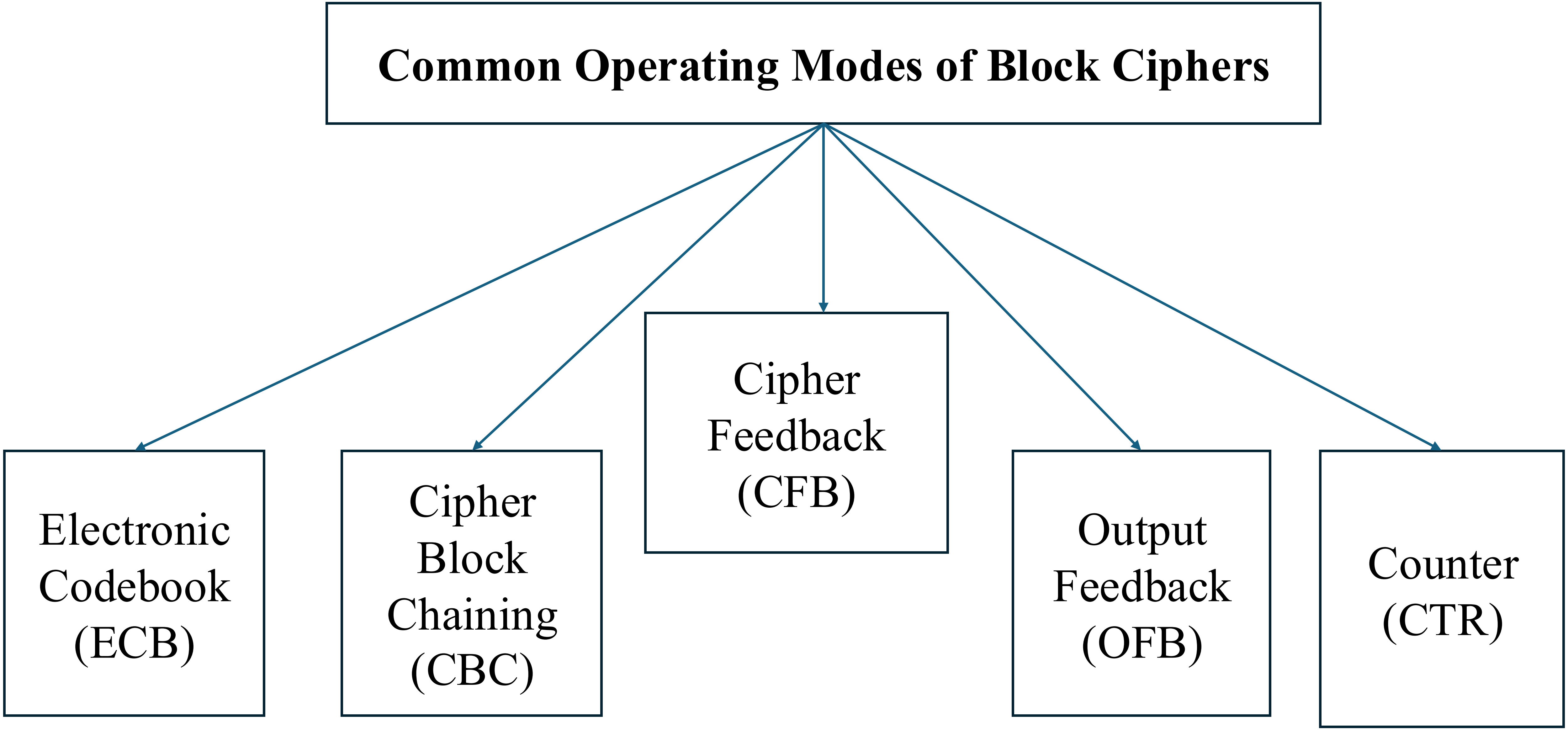
Triple DES with 3 Keys

- $C = E(K_3, D(K_2, E(K_1)))$
- $P = D(K_1, E(K_2, D(K_3)))$
- If $K_1 = K_3$ (same as Triple DES with 2 keys)
- If $K_1 = K_2 = K_3$ (same as DES)
- If K_1, K_2, and K_3 are different then highest level of security is offered.

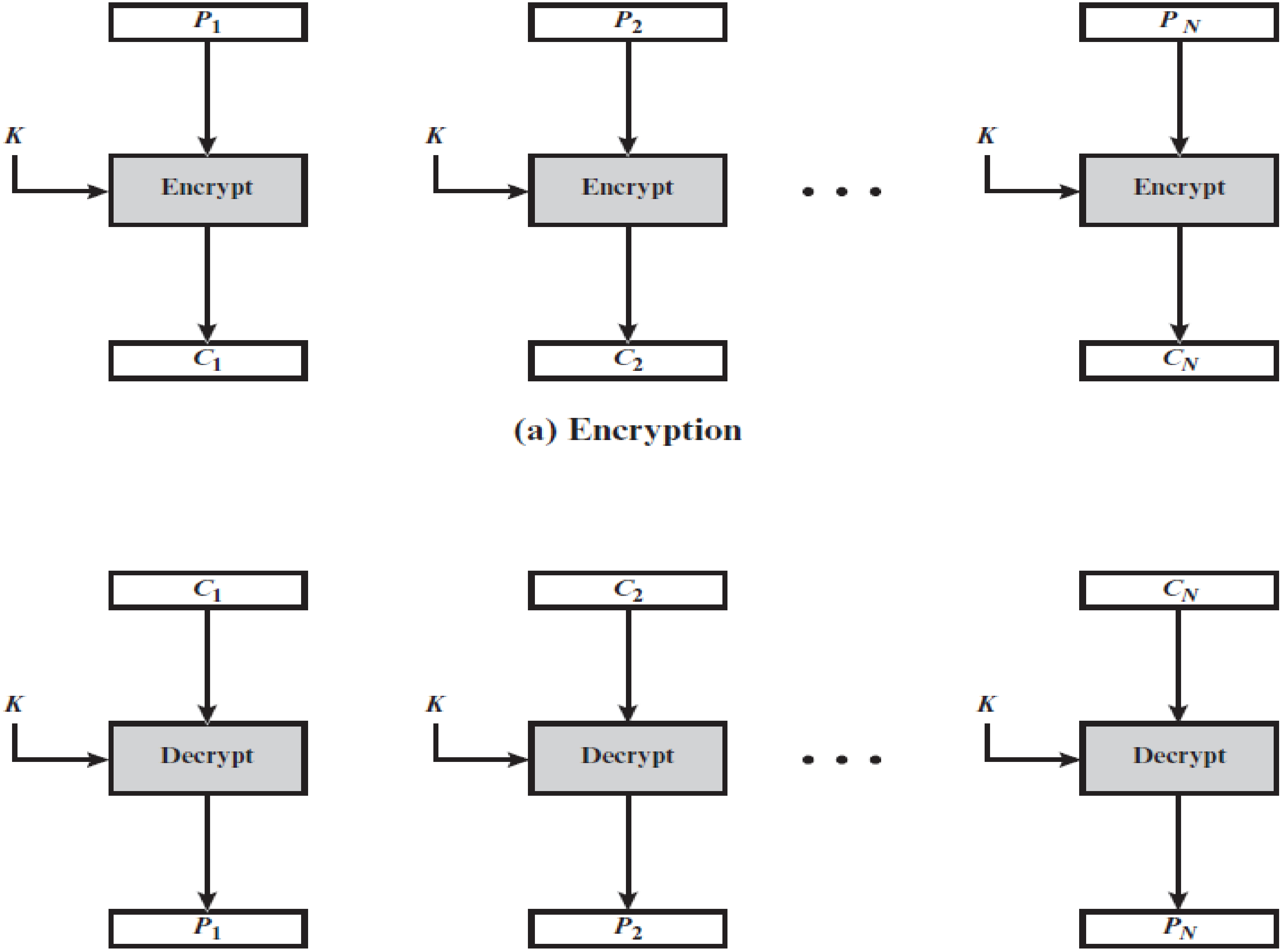
Triple DES with 3 Keys (Contd..)



COMMON OPERATING MODES OF BLOCK CIPHERS


Common Operating Modes of Block Ciphers
Electronic Codebook (ECB)
Cipher Block Chaining (CBC)
Cipher Feedback (CFB)
Output Feedback (OFB)
Counter (CTR)

ECB



(a) Encryption

(b) Decryption

ECB (Contd..)

- $C_i = E(K, P_i)$; where i = 1, 2, ……….., N.
- $P_i = D(K, C_i)$; where i = 1, 2, ……….., N.
- Each block is encrypted/decrypted using the same key producing corresponding CT/DT.
- A certain PT block will encrypt to the same CT block.

Pros and Cons of ECB

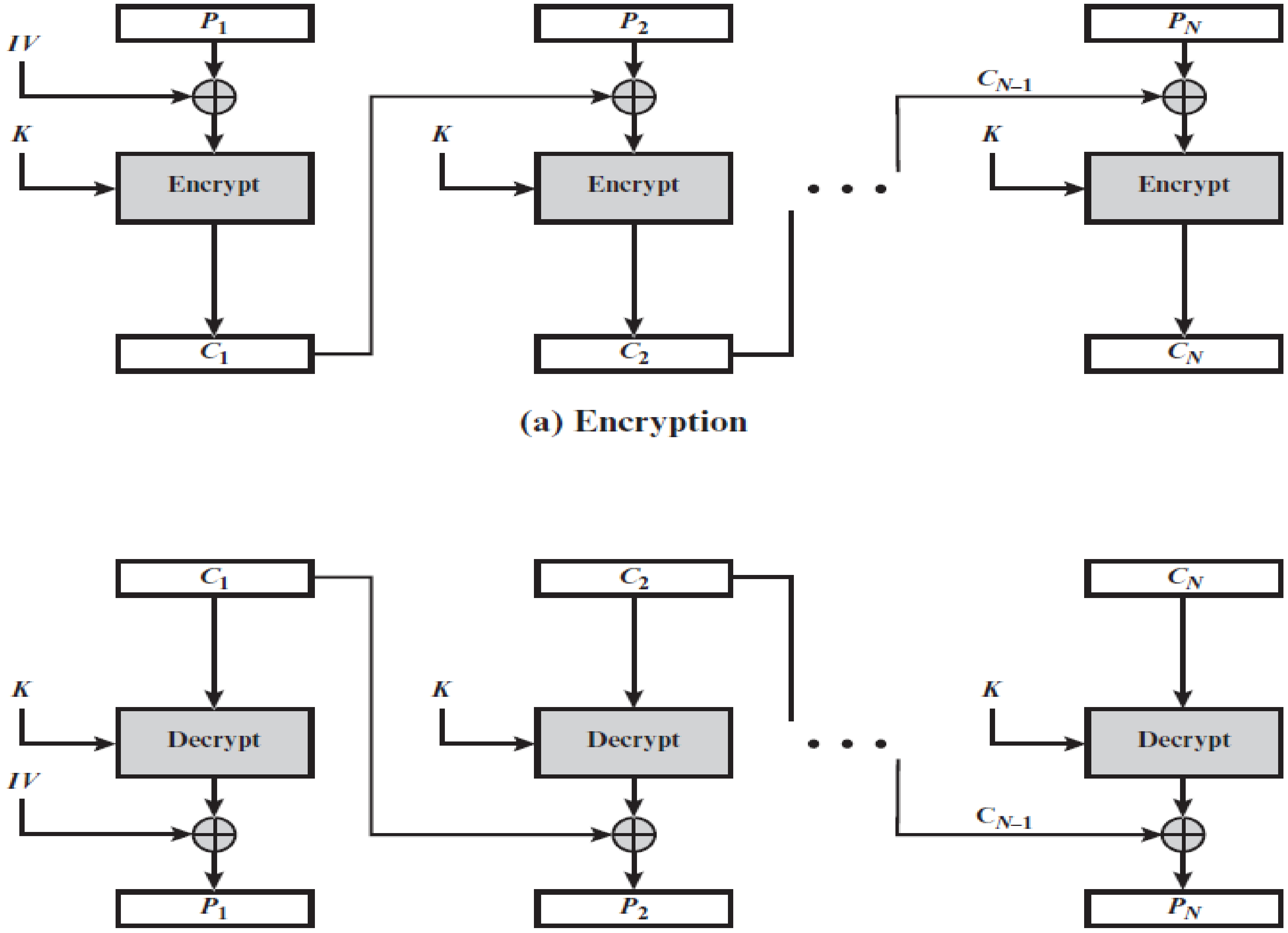
- Simple to understand and easy to implement.
- Parallel block encryptions/decryptions can be done, which eventually provides more efficiency.
- Error propagation doesn’t happen from one block to the subsequent blocks.

- Highly vulnerable to pattern attack (especially for large data).
- Provides poor diffusion.
- Padding might be required.

Properties and Criteria for designing modes superior to ECB

- Overhead
- Error Recovery
- Error Propagation
- Diffusion
- Security

CBC



(a) Encryption

(b) Decryption

CBC (Contd..)

- $C_1 = E(K, P_1 \oplus IV)$
- $C_i = E(K, P_i \oplus C_{i-1})$; where $i = 2, 3, 4, \ldots\ldots\ldots, N$

- $P_1 = D(K, C_1) \oplus IV$
- $P_i = D(K, C_i) \oplus C_{i-1}$; where $i = 2, 3, 4, \ldots\ldots\ldots\ldots\ldots, N$

- The size of IV is same as that of the blocks.
- The IV must be known and kept confidential to both sender and the receiver, and must be protected against unauthorized changes.
- IV must be unique for each session
- IV can be kept confidential through ECB encryption.
- Integrity of IV can be provided using Message Authentication Codes (MACs).

Threat to Integrity of IV in CBC

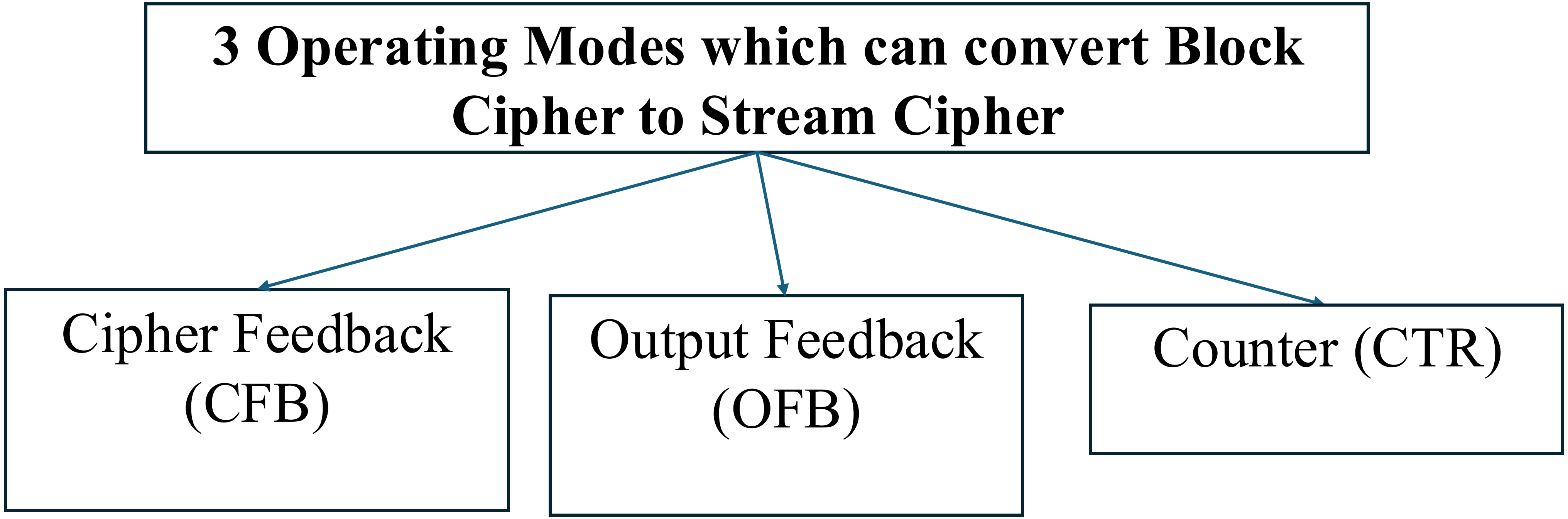
- $P_1 = IV \oplus D(K, C_1)$
- So, $P_1[i] = IV[i] \oplus D(K, C_1)[i]$
- Hence, $P_1[i]' = IV[i]' \oplus D(K, C_1)[i]$
- The attacker takes advantage of '$\oplus$' to manipulate the PT by altering IV.

- Recommended methods for unpredictable IV generations:- Nonce, Random Number Generator

Pros and Cons of CBC

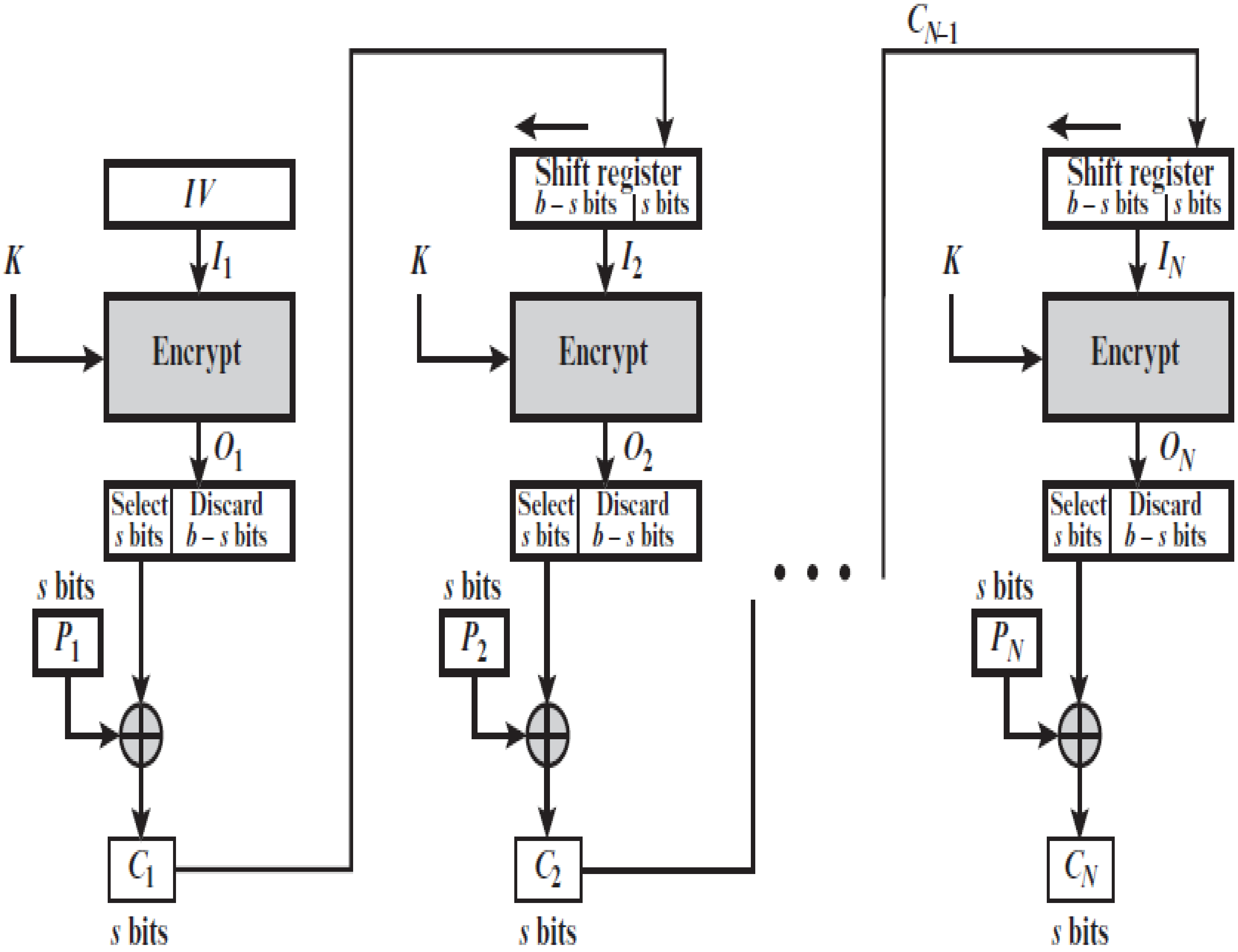
- Simple to understand and easy to implement.
- More resistant to Pattern attack when compared to that of ECB.
- Provides better diffusion property when compared to that of ECB.

- Requirement of secure IV generation for each session.
- CBC blocks are processed sequentially.
- Higher probability of error propagation when compared to that of ECB.
- Error Recovery rate is lower than that of ECB.

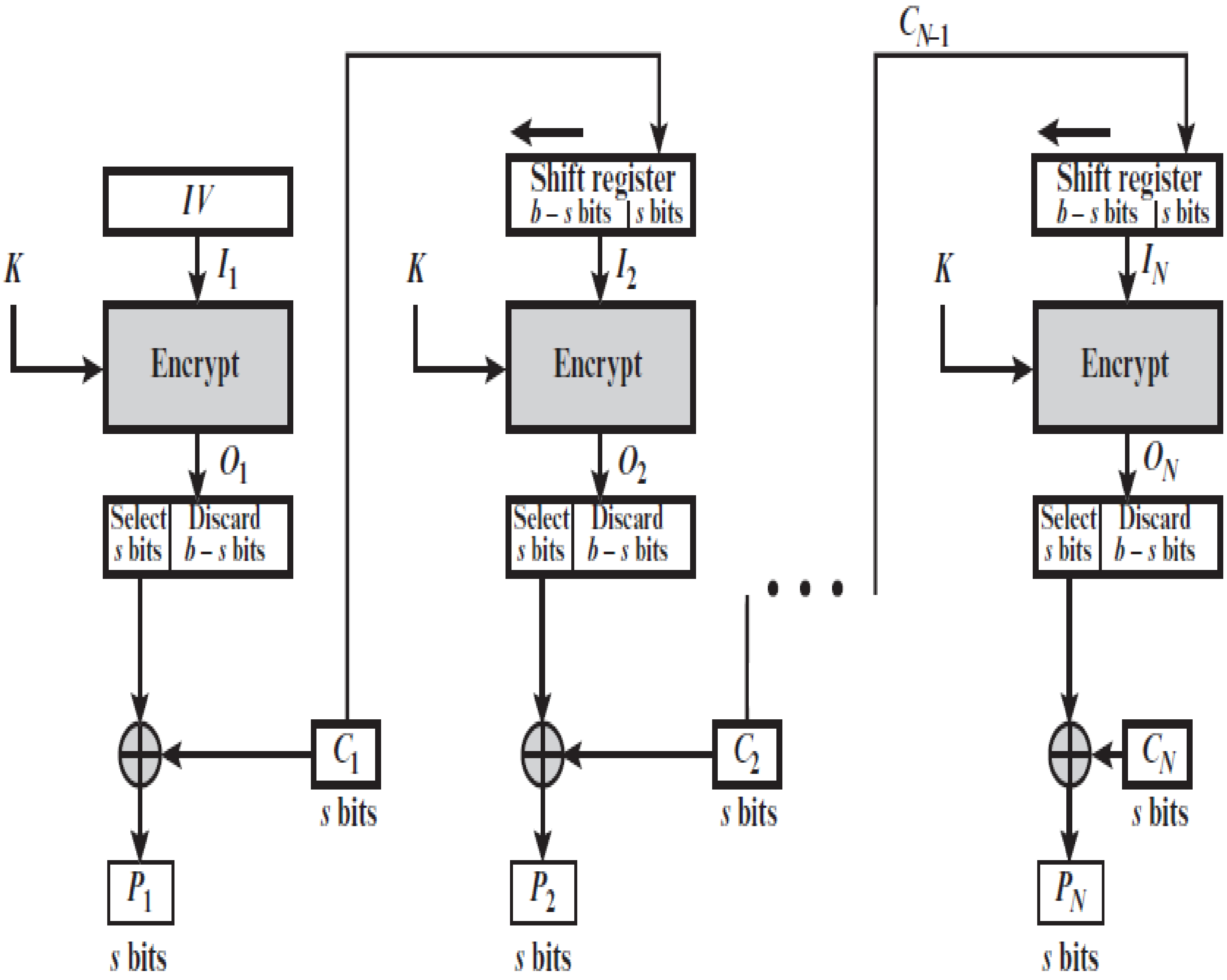


- Higher Efficiency
- Lower Latency
- Flexibility
- Error Propagation Control

CFB (Encryption)



CFB (Decryption)



CFB (Contd..)

- PT is divided into segments of 's' bits each.
- Popular choice of 's' = 1 Byte.

- $I_1 = IV$
- $I_j = LSB_{b-s}(I_{j-1}) || C_{j-1}$; $j = 2, 3, 4, \ldots\ldots\ldots, N$
- $O_j = E(K, I_j)$; $j = 1, 2, 3, \ldots\ldots\ldots\ldots, N$
- $C_j = P_j \oplus MSB_s(O_j)$; $j = 1, 2, 3, \ldots\ldots\ldots\ldots, N$

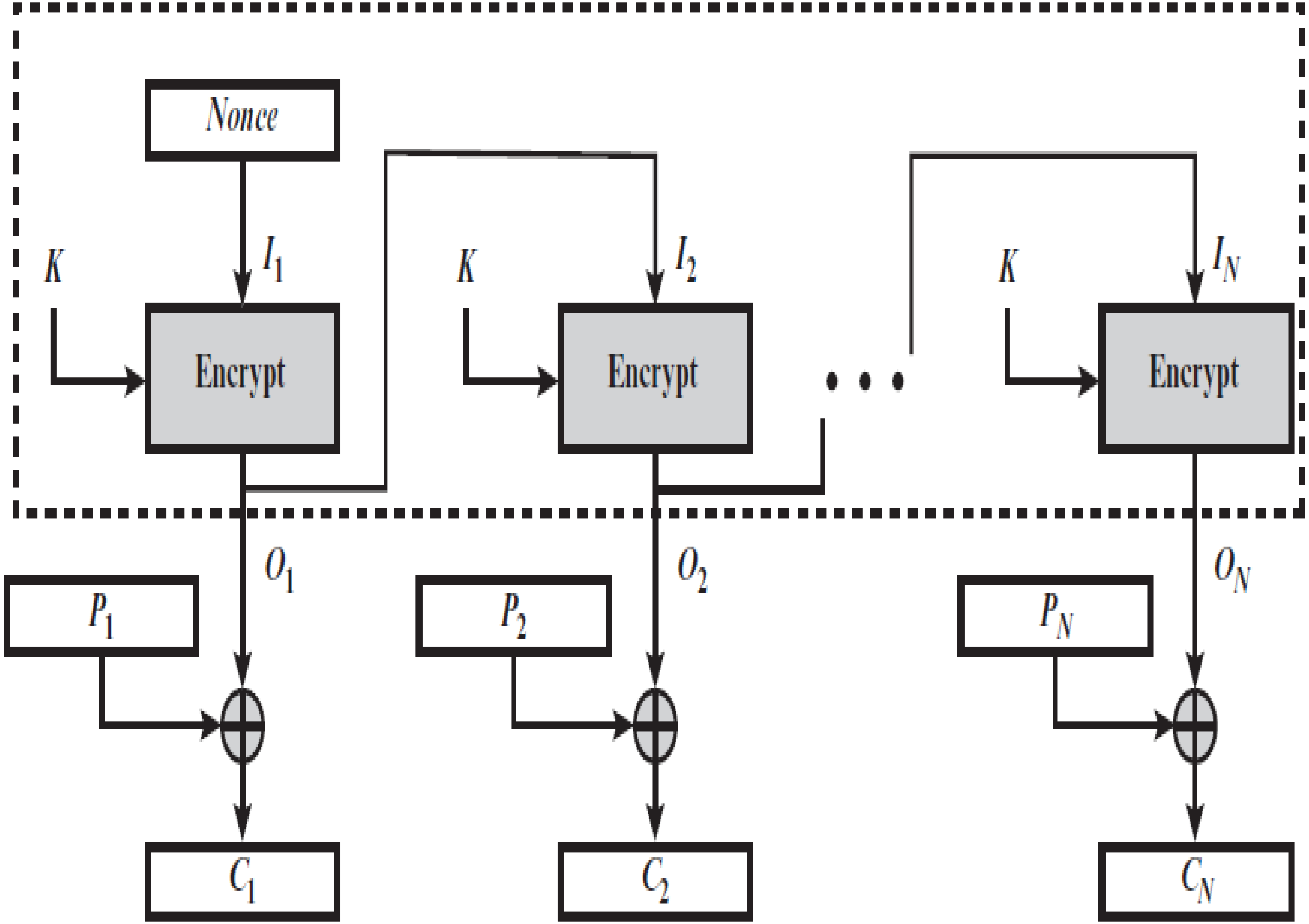
- $P_j = C_j \oplus MSB_s(O_j)$; $j = 1, 2, 3, \ldots\ldots\ldots\ldots, N$

Pros and Cons of CFB

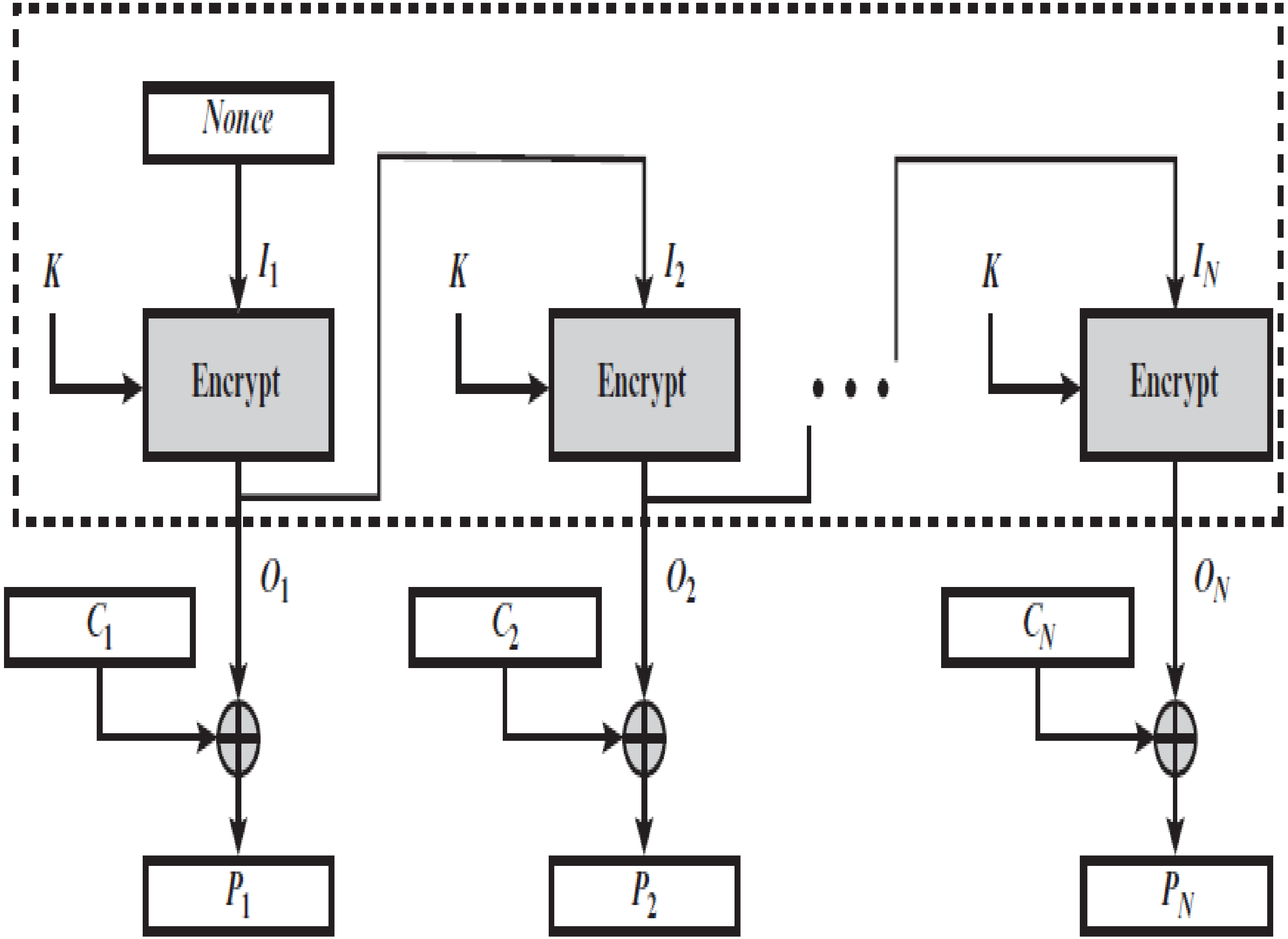
- Padding is not required.
- Encryption Function can be used for executing the corresponding Decryption function as well.
- Error Propagation is lesser compared to that of CBC.
- More Flexible.

- IV Management
- Limited Parallelism
- Not a typical stream cipher.
- Not suitable for encrypting or decrypting large data.

OFB (Encryption)



OFB (Decryption)



OFB (Contd..)

- Operates on full blocks ('b' bits each) of PT and CT.
- Size of Nonce is same as that of the blocks.

- $I_1 = \text{Nonce}$
- $I_j = O_{j-1}$; $j = 2, 3, \ldots\ldots\ldots\ldots\ldots, N$
- $O_j = E(K, I_j)$; $j = 1, 2, 3, \ldots\ldots\ldots, N$
- $C_j = P_j \oplus O_j$; $j = 1, 2, 3, \ldots\ldots\ldots, N - 1$
- $C_N = P_N \oplus MSB_u(O_N)$; where $u \le b$

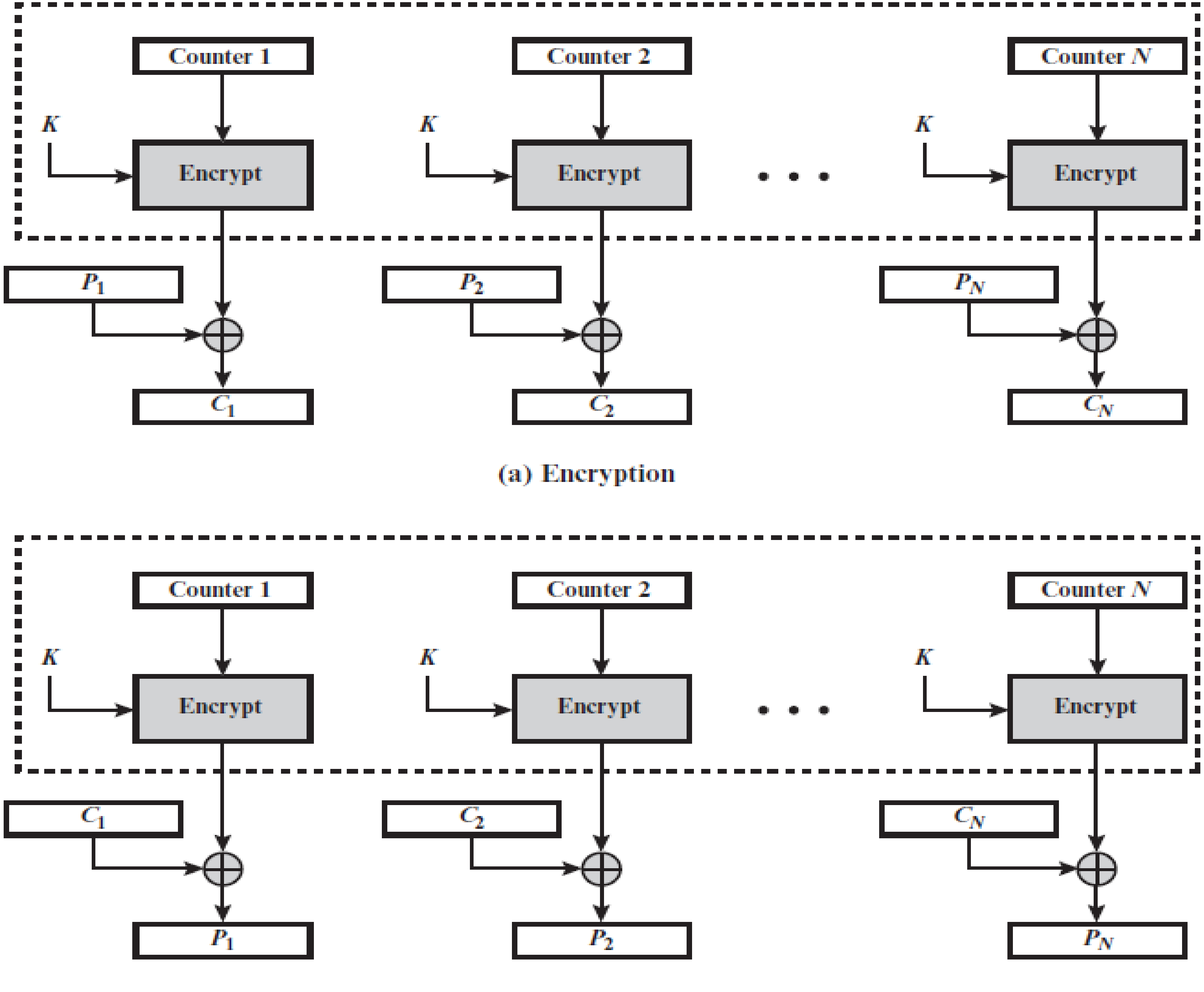
- $P_j = C_j \oplus O_j$; $j = 1, 2, 3, \ldots\ldots\ldots, N - 1$
- $P_N = C_N \oplus MSB_u(O_N)$; where $u \le b$

Pros and Cons of OFB

- Encryption Function can be used for executing the corresponding Decryption function as well.
- Error Propagation doesn’t happen.
- Provides partial parallel processing of the blocks.
- Padding is not required.

- IV Management
- More vulnerable to Message Stream Modification attack
- Provides a severe threat to Integrity.

CTR



(a) Encryption

(b) Decryption

CTR (Contd..)

- Size of the counter = Block size
- $T_j = (T_{j-1} + 1) \pmod{2^b}$

- $C_j = P_j \oplus E(K, T_j)$; $j = 1, 2, \ldots\ldots, N-1$
- $C_N = P_N \oplus MSB_u[E(K,T_N)]$; where $u <= b$

- $P_j = C_j \oplus E(K, T_j)$; $j = 1, 2, \ldots\ldots, N-1$
- $P_N = C_N \oplus MSB_u[E(K,T_N)]$; where $u <= b$

CTR (Pros and Cons)

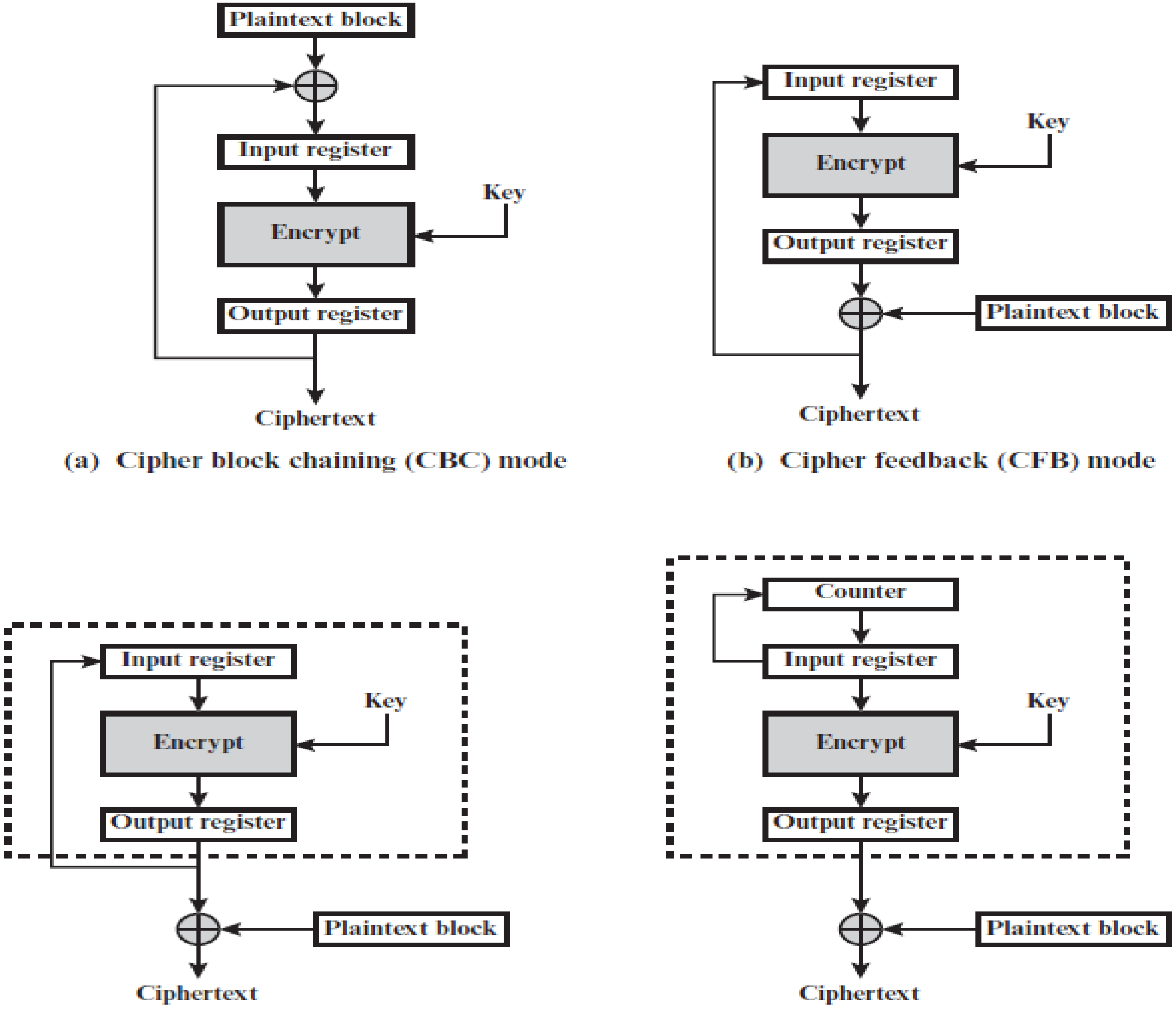
- Hardware Efficiency
- Software Efficiency
- Preprocessing
- Random Access
- Provable Security
- Simplicity

- Nonce Management
- Nonce reuse will leak information about the entire PT.
- Vulnerable to Message Stream Modification attack

Overview of Block Cipher Modes of Operation

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

Feedback Characteristics of Block Cipher Modes of Operation



(a) Cipher block chaining (CBC) mode

(b) Cipher feedback (CFB) mode

(c) Output feedback (OFB) mode

(d) Counter (CTR) mode

Feedback Characteristics of Block Cipher Modes of Operation

- Except ECB, the rest of the NIST approved modes involve feedback (FB).
- Regarded as the encryption function taking input from an input register (size equal to that of a block).
- The output of encryption is stored in an output register.
- The input register is updated one block at a time by FB mechanism.
- OFB and CTR produce encryption outputs independent of PT and CT (Hence natural candidates for stream cipher).

XTS-AES MODE FOR BLOCK-ORIENTED STORAGE DEVICES

Overview of XTS-AES

- XTS (XEX Tweakable Block Cipher with Ciphertext Stealing)
- Approved by NIST in 2010 as an additional mode.
- Defined by an IEEE standard (1619-2007) developed by P1619.
- Used for data encryptions on sector-based storage devices.
- Some of the applications are full-disk encryption, database encryption, secure cloud storage, etc.

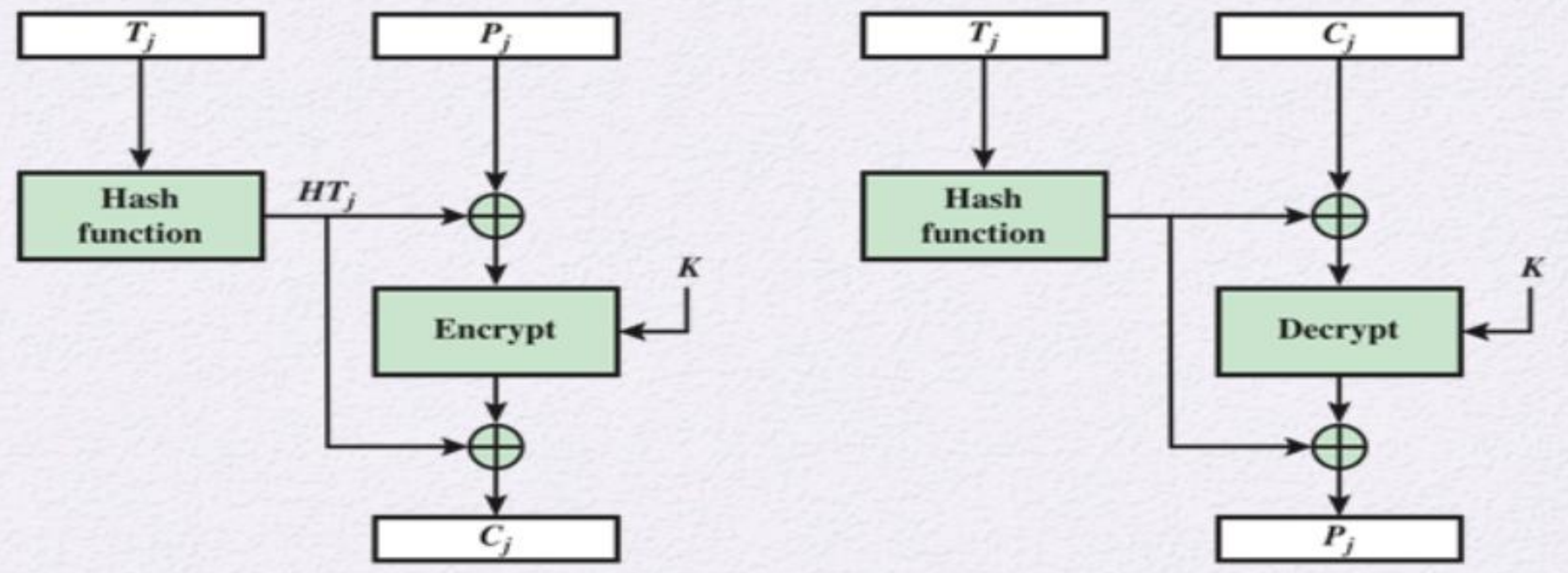
Key Points of XTS-AES

- Tweakable Block Ciphers
- Storage Encryption Requirements
- Operation on a Single Block
- Operation on a Sector

Tweakable Block Ciphers

- Foundation for XTS-AES.
- Has 3 inputs:- Plaintext (P), Symmetric Key (K), and a Tweak (T)
- ‘K’ is used to provide security, and ‘T’ is used to provide variability.

Tweakable Block Ciphers



(a) Encryption (a) Decryption

Tweakable Block Ciphers

- $h = H(T)$
- $C = h \oplus E(K, h \oplus P)$
- $P = h \oplus D(K, h \oplus C)$

- Use of different T with same K and same P would produce different Ciphertexts.
- Use of tweaks makes it easier to construct any operating mode.

Tweakable Block Ciphers (Pros and Cons)

- Enhances the security of any operating mode.
- Versatile
- Provides Integrity of data
- Key Management gets easier

- Management of Tweaks
- The Cipher is more vulnerable if the Tweak space is small

Storage Encryption Requirements (Defined by P1619)

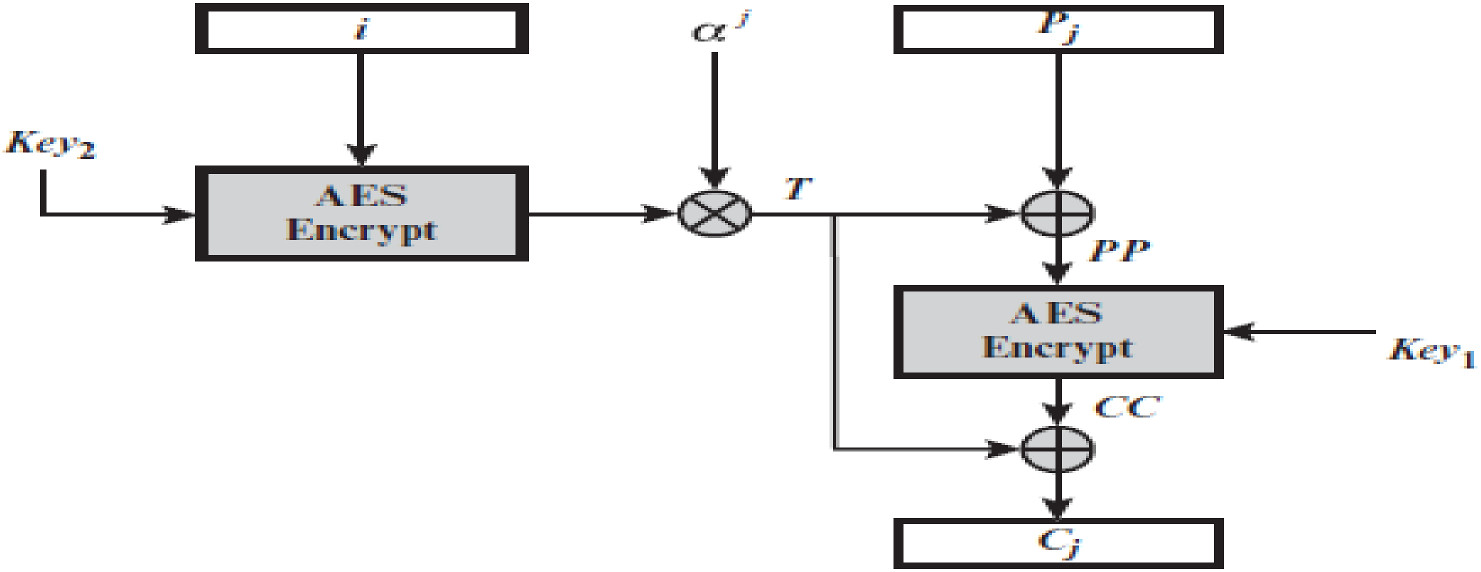
- The ciphertext is freely available for an attacker.
- The data layout is not changed on the storage medium and in transit.
- Data are accessed in fixed sized blocks, independently from each other.
- Encryption is performed in 16-byte blocks, independently from other blocks.
- There are no other metadata used, except the location of the data blocks within the whole data set.
- The same plaintext is encrypted to different ciphertexts at different locations, but always to the same ciphertext when written to the same location again.
- A standard conformant device can be constructed for decryption of data encrypted by another standard conformant device.

Vulnerabilities identified by P1619 group for stored data encryptions by traditional modes

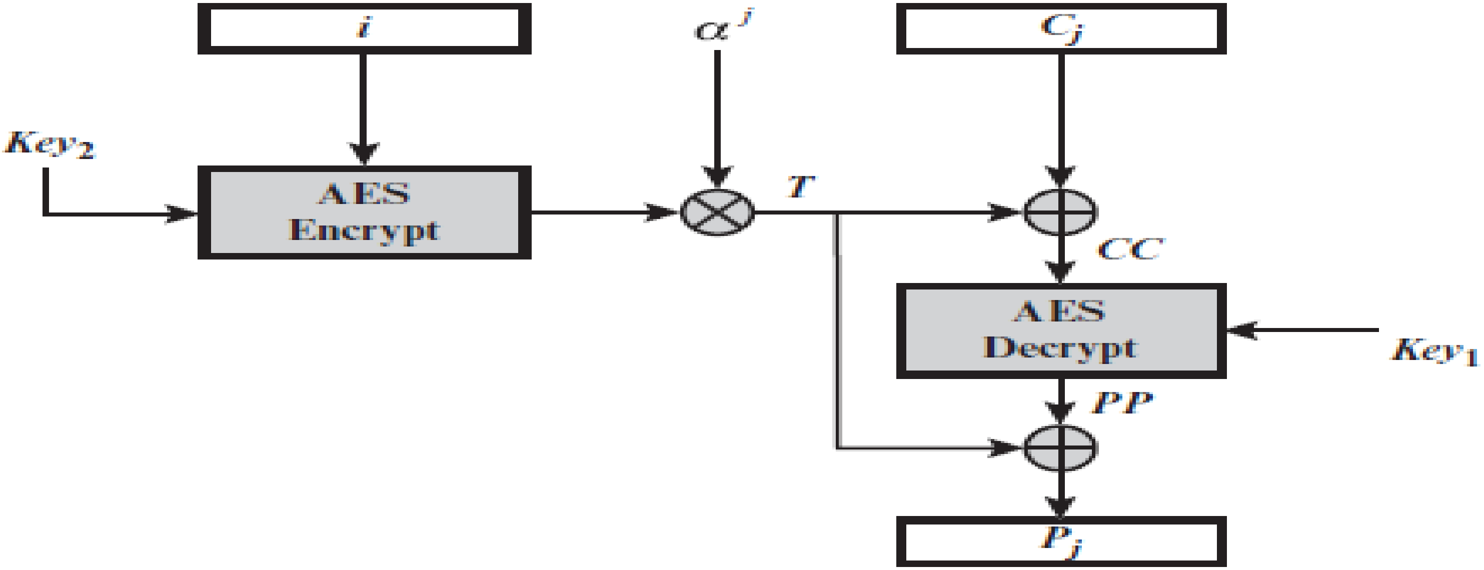
- IV Prediction attack in CBC
- CT copying in CBC
- Bit Flipping attack in CBC.

- Bit Flipping attack in CTR
- Counter Synchronization Issues in CTR
- Predictable Counter values attack in CTR

XTS-AES Operation on a Single Block



(a) Encryption



(b) Decryption

- $GF(2^{128}) \rightarrow (x^{128} + x^7 + x^2 + x + 1)$
- $\alpha \rightarrow x$

XTS-AES Operation on Single Block (Contd..)

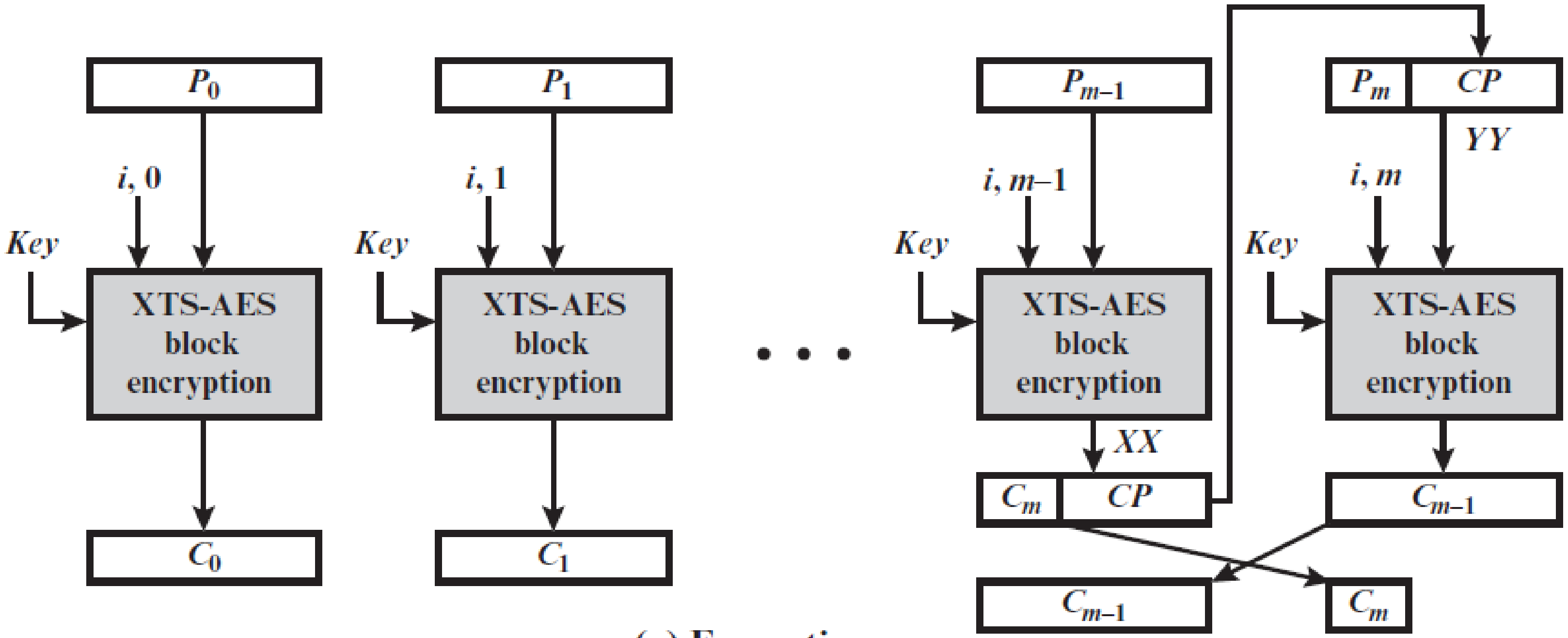
Encryption:-

- $T = E(K_2, i) \otimes \alpha^j$
- $PP = P \oplus T$
- $CC = E(K_1, PP)$
- $C = CC \oplus T$

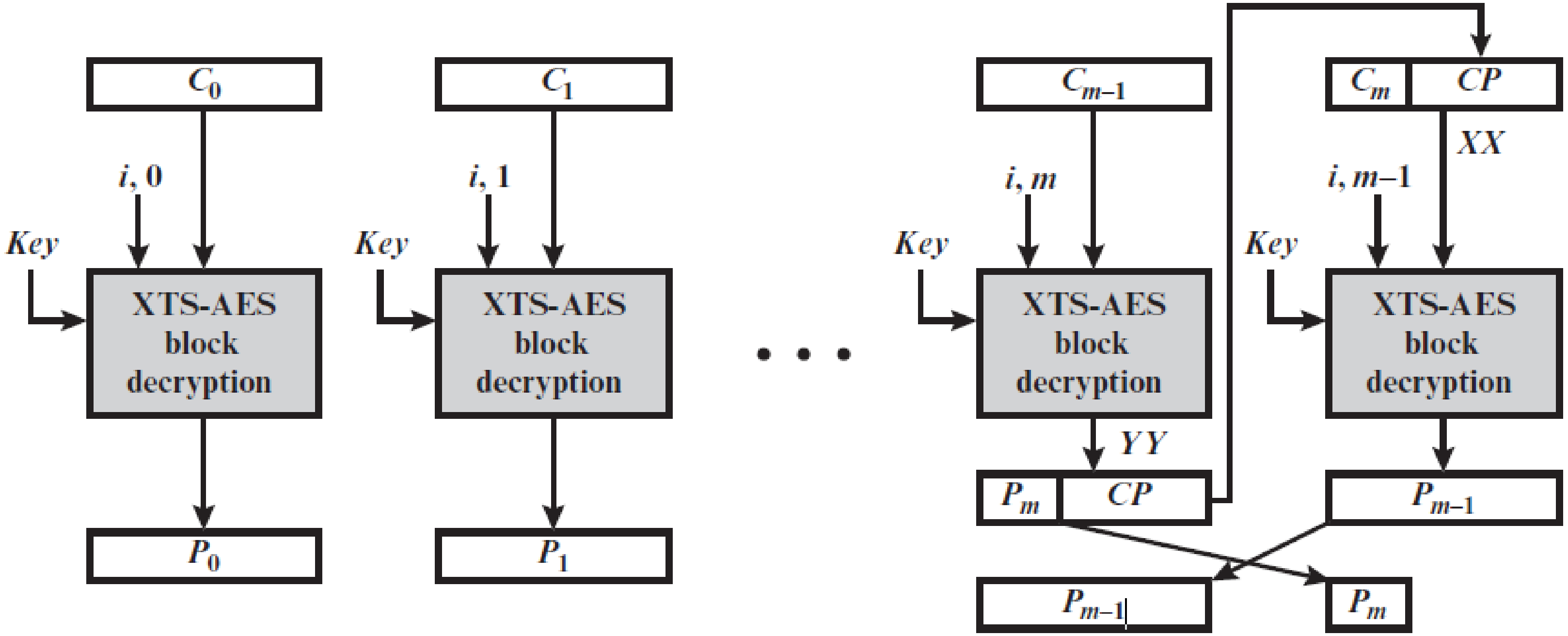
Decryption:-

- $CC = C \oplus T$
- $PP = D(K_1, CC)$
- $P = PP \oplus T$

XTS-AES Operation on a Sector



(a) Encryption



(b) Decryption

XTS-AES Operation on a Sector (Contd..)

- PT $\rightarrow$ (P_0, P_1, P_2, ……., P_{m-1}, P_m; 128 bits each till $(m-1)^{th}$ block)
- P_m ('s' bits); where $1 \leq s \leq 127$ bits.
- C_{m-1} is the last CT block having 128 bits.

<u>XTS-AES mode with null final block:-</u>

- C_j = XTS-AES-blockEnc(K, P_j, i, j); j = 0, 1, ……., m-1
- P_j = XTS-AES-blockDec(K, C_j, i, j); j = 0, 1, ……., m-1

XTS-AES Operation on a Sector, when final block is incomplete (Encryption)

- C_j = XTS-AES-blockEnc(K, P_j, i, j); j= 0, 1, ……., m-2
- XX = XTS-AES-blockEnc(K, P_{m-1}, i, m-1)
- CP = LSB_{128-s}(XX)
- YY = $P_m \| CP$
- C_{m-1} = XTS-AES-blockEnc(K, YY, i, m)
- C_m = MSB_s(XX)

XTS-AES Operation on a Sector, when final block is incomplete (Decryption)

- P_j = XTS-AES-blockDec(K, C_j, i, j); j= 0, 1, ……., m-2
- YY = XTS-AES-blockDec(K, C_{m-1}, i, m-1)
- CP = LSB_{128-s}(YY)
- XX = $C_m \| CP$
- P_{m-1} = XTS-AES-blockDec(K, XX, i, m)
- P_m = MSB_s(YY)

Pros and Cons of XTS-AES

- Parallel Processing (Except the last incomplete block)
- Flexibility
- More secure compared to the traditional modes of operations of Block Ciphers.
- Suits well for confidentiality in sector-based storage devices.
- Provides a minor level of Data Integrity.

- Complex Implementation
- Generally limited to data at rest.
- Generally, doesn’t suite for a network of devices.
- Key Management Issues