# Cryptography and Security

# Lecture 9

# Block Cipher Operation

# Multiple Encryption and DES

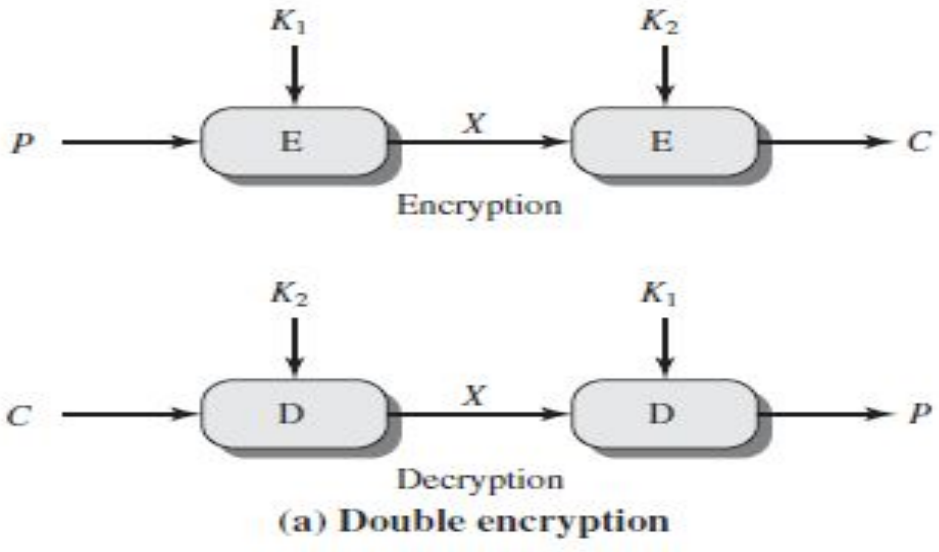
- A replacement for DES was needed
  - demonstrated exhaustive key search attacks
- AES is a new cipher alternative
- Prior to this alternative was to use multiple encryption with DES implementations
- Triple-DES is the chosen form

# Double DES

- Given a plaintext ***P***, two encryption key $K_1$ and $K_2$, ciphertext ***C*** is generated as

$$C = E(K_2, E(K_1, P))$$

- Decryption requires the keys be applied in reverse order:



(a) Double encryption

# Issues with Double DES

- Assumption of $E(K_2, E(K_1, P)) = E(K_3, P)$
  - With $2^{64}$ possible inputs, the number of mappings that generate a permutation of the input blocks is $(2^{64})! > 10^{1020}$
  - DES defines one mapping for each different key, thus DES uses $2^{56}$ mappings.
  - If DES is used twice with different keys, it will produce one of the many mappings that are not defined by a single application of DES.

# Issues with Double DES

- **Meet-in-the-middle attack**: Based on the assumption

  $C = E(K_2, E(K_1, P))$

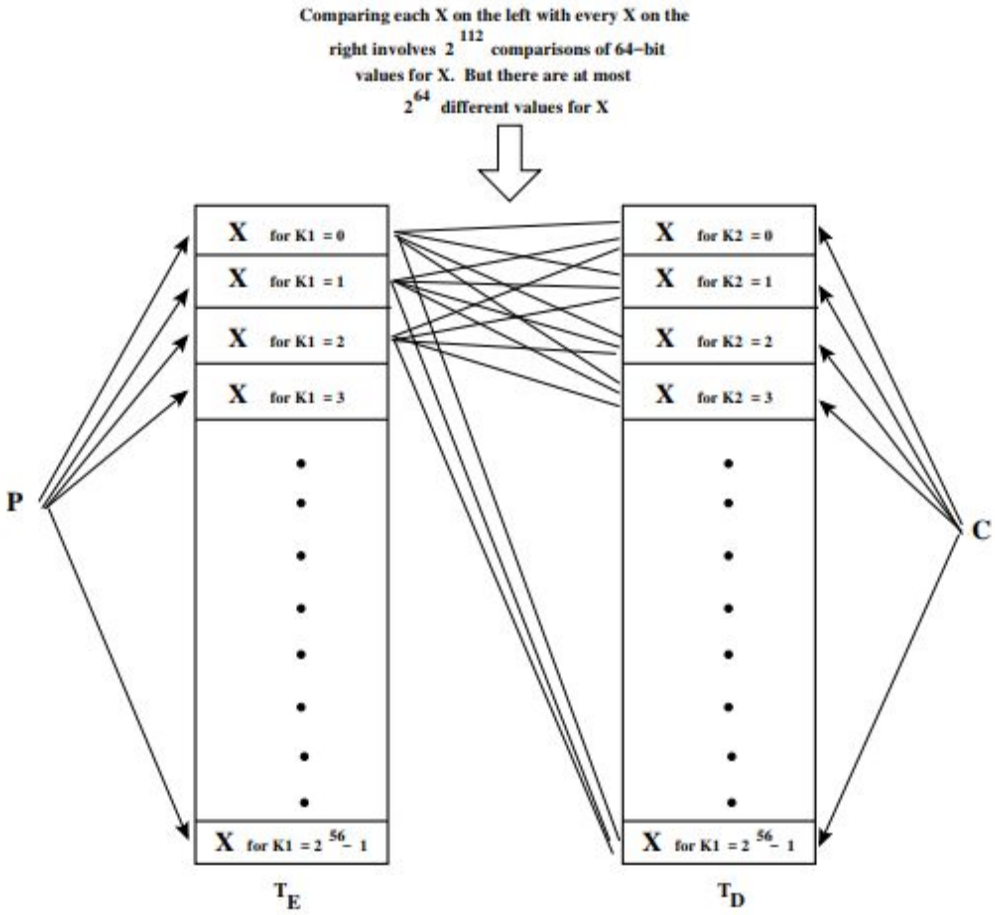
  $X = E(K_1, P) = D(K_2, C)$

- Given a pair ***(P, C)***, the attack proceeds as follows.
  - Encrypt ***P*** for all possible $2^{56}$ values of $K_1$ and stores the result in a table.
  - Sort the table by the value of ***X***.
  - Decrypt ***C*** using all possible $2^{56}$ values of $K_2$ and for each decryption, check the result against the table for a match.

# Issues with Double DES

- If a match occurs, then test $(K_1, K_2)$ for a new plaintext-ciphertext pair.
- If the two keys produce the correct ciphertext, accept them as the correct keys.

# Computational Complexity of Meet-in-the Middle Attack (for 2DES)




- $2^{112}$ comparisons are required to determine which entries are equal. This involves $2^{64}$ values of X. Thus $2^{112}/2^{64}=2^{48}$ comparisons must involves identical values.
- Therefore, when comparing entries of two tables number of false alarms is $2^{48}$.
- For another pair (P', C') we construct $T_{E'}$ and $T_{D'}$ each having $2^{48}$ entries. So the number of false alarm is $2^{48}/2^{64}=2^{-16}$.
- Thus the probability of getting a single pair $(K_1, K_2)$ that is correct key is $1-2^{-16}$.
- The effort required to make such a comparison is proportional to the size of $T_E$ and $T_D$, which is $2^{56}$, which is comparable to the effort required to break the single DES.

# Triple DES with Two Keys

- Defense to the ***meet-in-the-middle*** attack.
- $C = E(K_3, D(K_2, E(K_1, P)))$

  $P = D(K_1, E(K_2, D(K_3, C)))$
- Key length = 168 bits which may be unwieldy for some applications.
- Used in a number of Internet-based applications such as PGP and S/MIME.

# Triple DES with Two Keys

- The function follows an encrypt-decrypt-encrypt (EDE) sequence:

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

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

- Use of decryption allows users of 3DES to decrypt data encrypted by the users of the older single DES:

$$C = E(K_1, D(K_1, E(K_1, P))) = E(K_1, P)$$

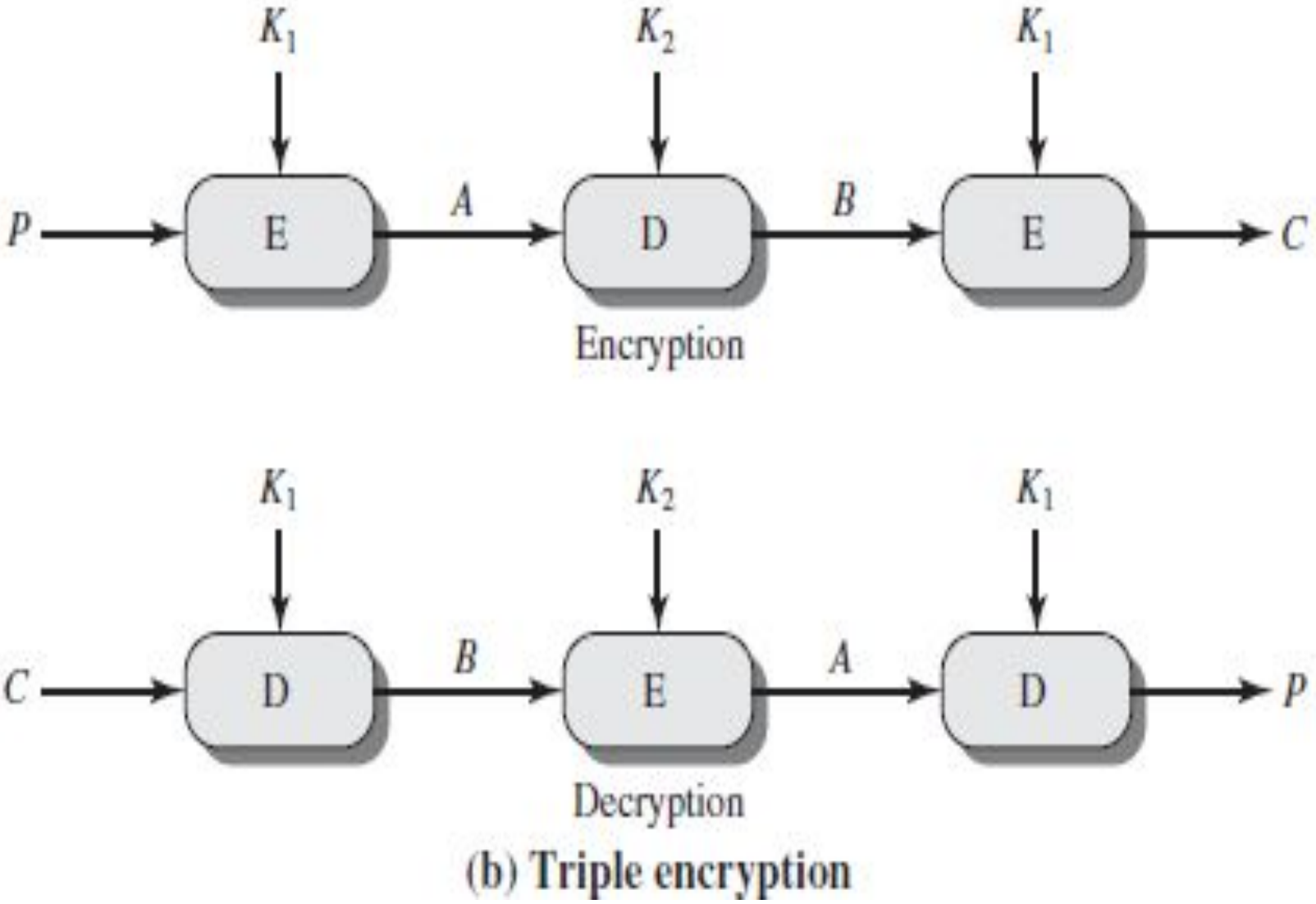
$$P = D(K_1, E(K_1, D(K_1, C))) = D(K_1, C)$$

- 3DES with two keys is a relatively popular alternatives to DES and has been adopted in key management standards ANSI X9.17 and ISO 8732.

# Triple DES with Two Keys

- A decryption stage between two encryption stages does not weaken the resulting cryptographic system in any way.
  - decryption in DES works in exactly the same manner as encryption. So if you encrypt data with one key and try to decrypt with a different key, the final output will be still be an encrypted version of the original input.

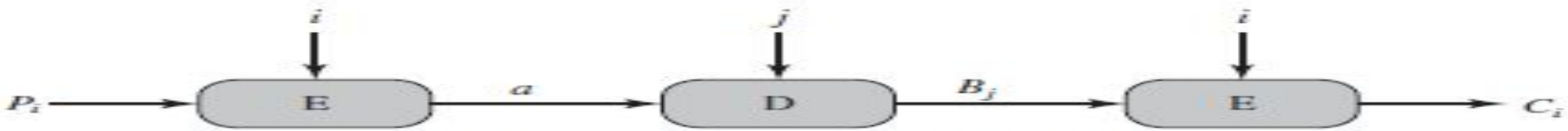
# Triple DES with Two Keys



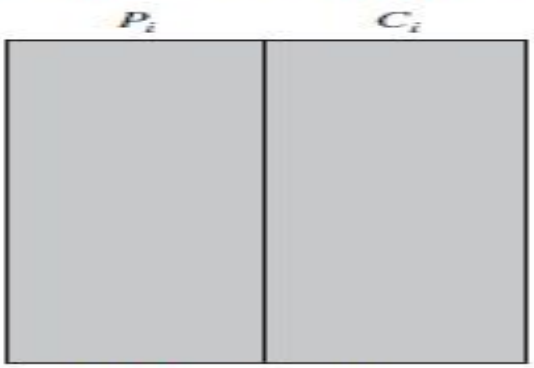

(b) Triple encryption

# Attacks on Triple DES with two keys

- It is theoretically possible to extend the meet-in-the-middle attack to 3DES based on two keys.
- If the attacker gets the access of the intermediate value ***A*** for a given plaintext ***P***, breaking the 3DES cipher becomes the same as breaking 2DES with the meet-in-the-middle attack.




(a) Two-key triple encryption with candidate pair of keys




(b) Table of $n$ known plaintext-ciphertext pairs, sorted on $P$


(c) Table of intermediate values and candidate keys

Figure 6.2 Known-Plaintext Attack on Triple DES

# Attacks on Triple DES with two keys

- **Step 1:** The attacker procures ***n*** pairs of ***(P, C)***. These are arranged in a two-column table, with all the ***P***'s in one column and their corresponding ***C***'s in the other column.
- **Step 2:** The attacker now chooses an arbitrary value ***a*** for ***A***. The attacker figures out the plaintext that will result in ***a*** for every possible key $K_1 = i: P_i = D(i, a)$

  If a $P_i$ matched in Table I, creates an entry in Table II consisting of value of $K_1$ and ***B*** such that $B = D(i, C)$ for ***C*** that corresponds to $P_i$.

  Sort Table II by ***B***.
- **Step 3:** For $K_1$, search for each $2^{56}$ possible $K_2$ using $B_j = D(j, a)$. Search $B_j$ in Table II and if there is a match then $(i, j)$ is the candidate value for $(K_1, K_2)$.
- **Step 4:** Test for other few plaintext-ciphertext pairs. If $(K_1, K_2)$ produces the desired ciphertext, the task is complete.

# Attacks on Triple DES with two keys

- For a given ***(P, C)***, the probability of selecting unique ***a*** that leads to success is $1/2^{64}$.
- Thus given ***n (P, C)*** pairs, the probability of success for a single selected value of ***a*** is $n/2^{64}$.
- The expected number of draws required to draw one red ball from a bin containing ***n*** red balls and ***N − n*** green balls is ***(N + 1)/(n + 1)*** if the balls are NOT replaced.
- So given the ***n*** pairs for ***(P, C)***, the number of different possible values for ***a*** that we may have to try is

$$\frac{2^{64}+1}{n+1} \approx \frac{2^{64}}{n}$$

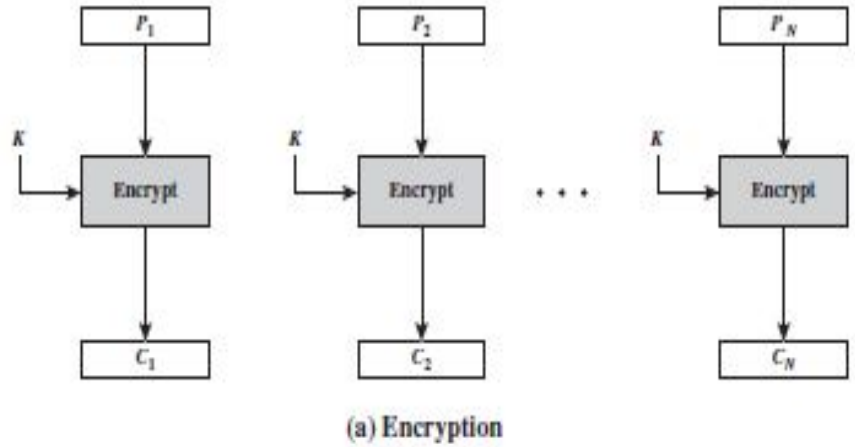
- The expected running time of the attack is on the order of

$$(2^{56})\frac{2^{64}}{n} = 2^{120-\log_2 n}$$

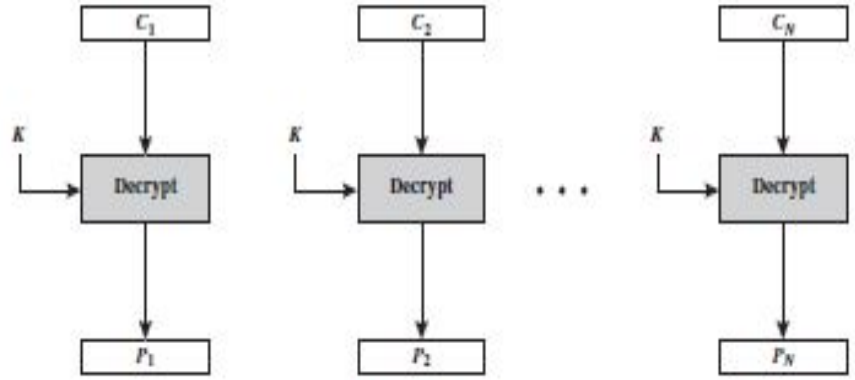
# Block Cipher Modes of Operation

- A block cipher takes a fixed-length block of text of length ***b*** and a key as input and produces a ***b***-bit block of ciphertext.
- If the amount of plaintext is greater than ***b*** bits, the plaintext is divided into blocks of ***b*** bits.
- When multiple blocks of plaintext is encrypted with same key, a number of security issues arises.
- To apply a block cipher in a variety of applications, five modes of operation have been defined:
  - Electronic Code Book
  - Cipher Block Chaining
  - Cipher Feedback
  - Output Feedback
  - Counter (CTR)

# Electronic Code Book




(a) Encryption



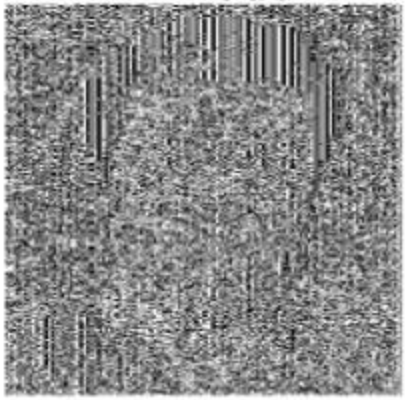

(b) Decryption

- Ideal for short amount of data, such as to transmit AES and DES key.
- If same b-bit blocks of plaintext appears more than once in the message, it always produce the same ciphertext.
- For lengthy messages, if the message is highly structured, it may be possible to exploit these regularities.

# Electronic Code Book

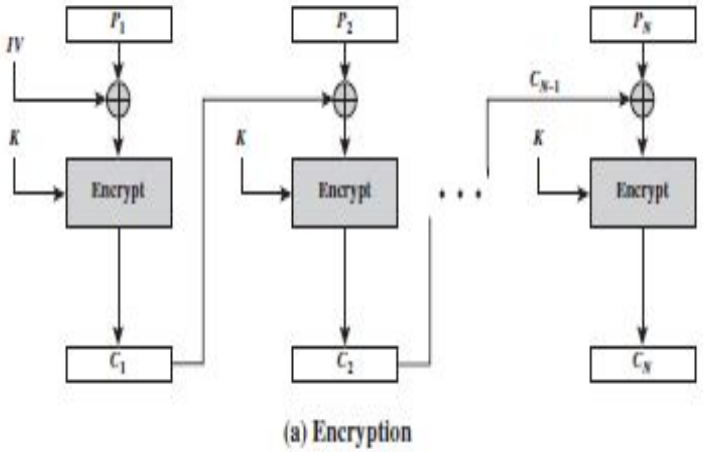
An example plaintext

Encrypted with AES in ECB mode

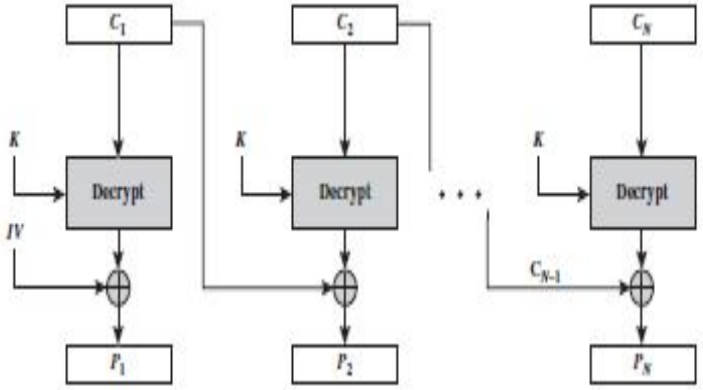


Courtesy: B. Preneel

# Cipher Block Chaining Mode




(a) Encryption




(b) Decryption

Figure 6.4 Cipher Block Chaining (CFB) Mode

- Overcome the security deficiencies of ECB.

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

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

- IV must be protected against unauthorized changes. This could be done by sending IV using ECB encryption.

# Cipher Block Chaining Mode

- If an opponent is able to fool the receiver into using a different value for IV, the opponent is able to invert selected bits in the first block of plaintext.

$$C_1 = E(K, [IV \oplus P_1])$$
$$P_1 = IV \oplus D(K, C_1)$$

$$P_1[i] = IV[i] \oplus D(K, C_1)[i]$$

$$P_1[i]' = IV[i]' \oplus D(K, C_1)[i]$$

# Cipher Block Chaining Mode

- IV can be generated in two ways:
  - Nonce: unique value unique for each execution of encryption algorithm. May be a counter, a timestamp or a message number.
  - Generate a random block using a random number generator.
- Applications:
  - Suitable for encrypting messages of length greater than b bits.
  - Used for achieving confidentiality and authentication.

# Cipher Feedback Mode

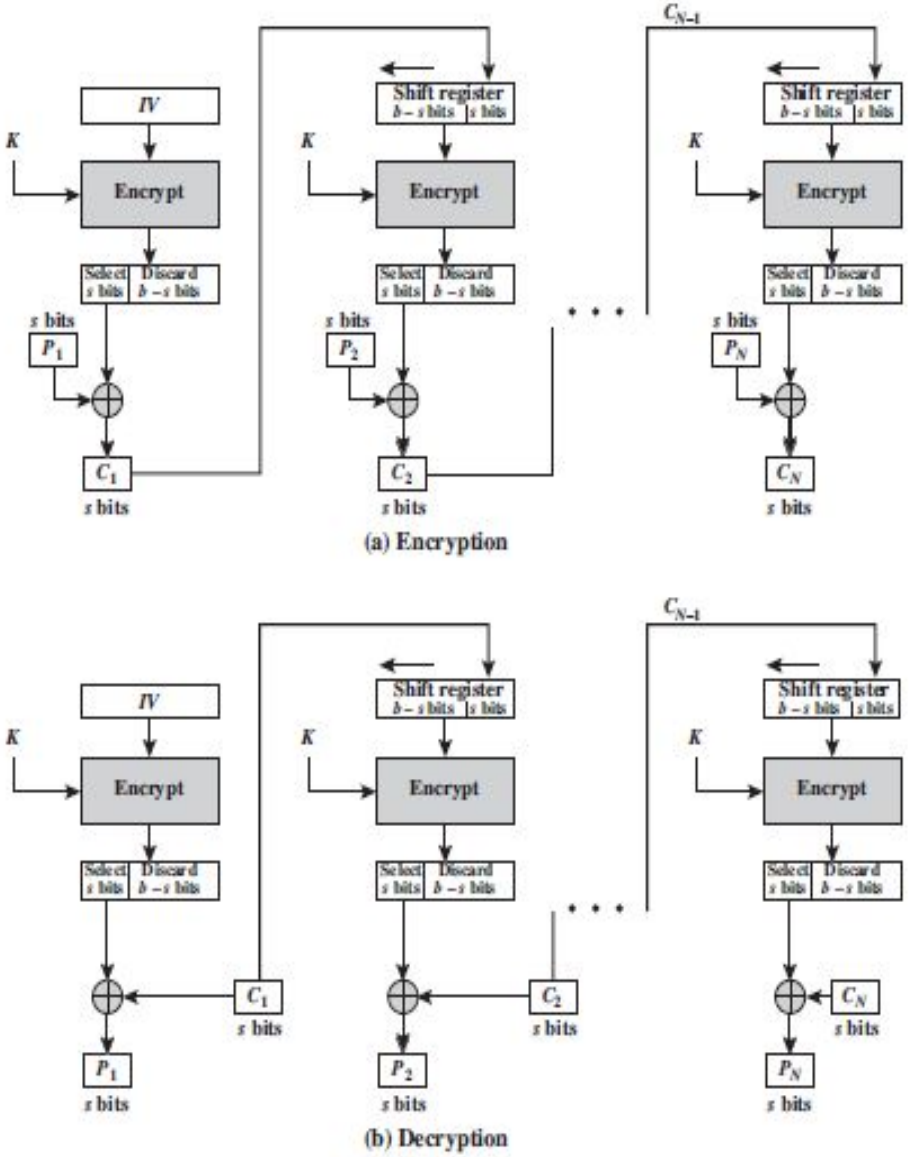



Figure 6.5 s-bit Cipher Feedback (CFB) Mode

- message is treated as a stream of bits.
  - Eliminates the need to pad a message to be an integral number of blocks.
  - If a character stream is being transmitted, each character can be encrypted and transmitted immediately using a character-oriented stream cipher.

# Cipher Feedback Mode

- Encryption function is used in decryption.

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

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

- In encryption, each forward cipher function depends on the result of the previous forward cipher function. Thus multiple forward cipher operations cannot be performed in parallel.
- In decryption, the forward cipher functions can be performed in parallel if the input blocks are first constructed from the IV and ciphertext.

# Output Feedback Mode

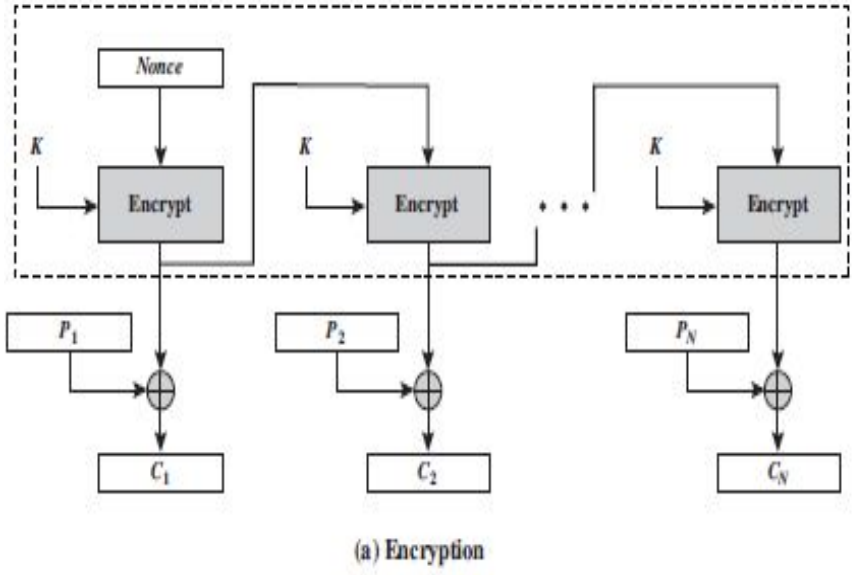



Figure 6.6 Output Feedback (OFB) Mode

- $C_j = P_j \text{ XOR } E(K, O_{j-1})$

  $O_{j-1} = E(K, O_{j-2})$
- $C_j = P_j \text{ XOR } E(K, C_{j-1} \text{ XOR } P_{j-1})$

  $P_j = C_j \text{ XOR } E(K, C_{j-1} \text{ XOR } P_{j-1})$
- If the size of the last block is $u<b$ bits, the most significant $u$ bits of the last output block $O_N$ are used for XOR operation. Remaining bits of the last output block are discarded.
- In OFB, the IV must be a nonce which is unique for each execution of the encryption operation. Otherwise, identical plaintext at identical position of two different message will produce same ciphertext.

# Output Feedback Mode

- In OFB, bit errors in transmission do not propagate. If bit error occurs in $C_1$, the recovered value of $P_1$ is affected.
- Vulnerable to message stream modification attack. Changing one bit in the ciphertext directly affect the corresponding bit in the plaintext. It is possible for opponent to make changes to some portion of data and then make changes to checksum. The receiver cannot detect the changes in the plaintext.

# Counter Mode

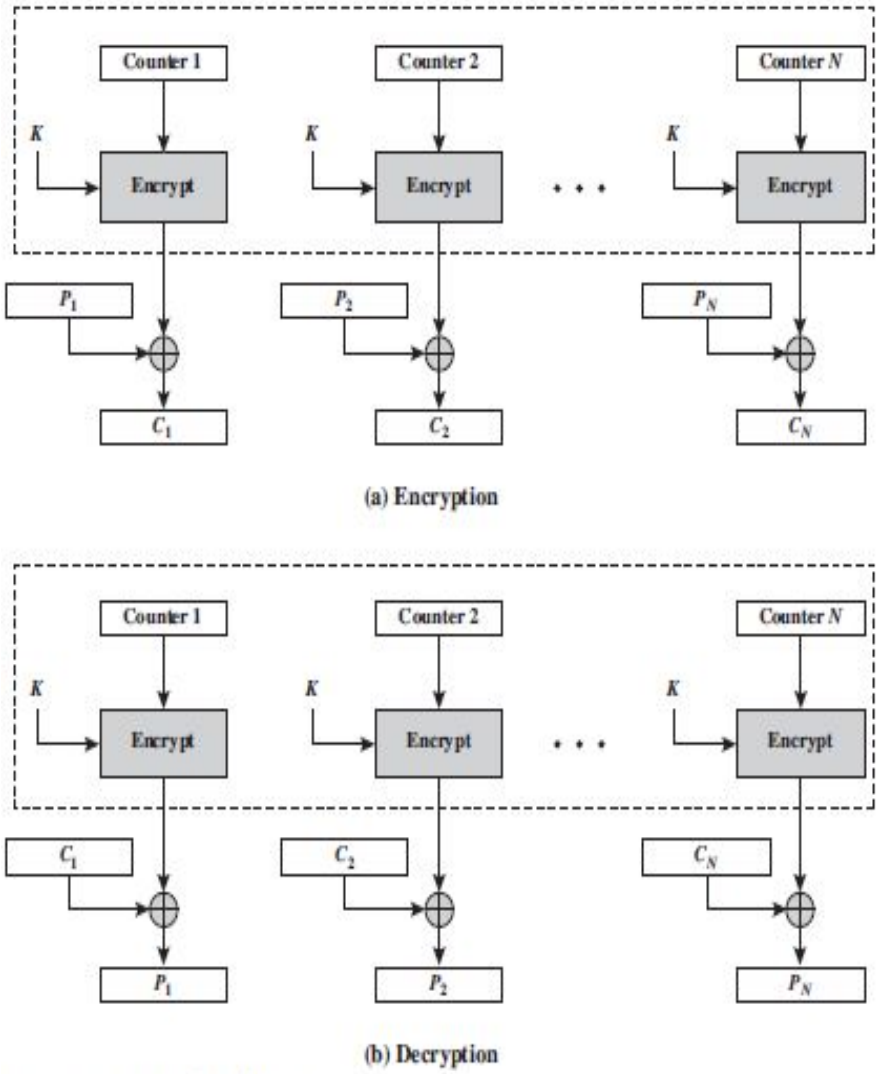



Figure 6.7 Counter (CTR) Mode

| | | |
|---|---|---|
| CTR | $C_j = P_j \oplus E(K, T_j) \quad j = 1, \ldots, N-1$<br>$C_N^* = P_N^* \oplus \text{MSB}_u[E(K, T_N)]$ | $P_j = C_j \oplus E(K, T_j) \quad j = 1, \ldots, N-1$<br>$P_N^* = C_N^* \oplus \text{MSB}_u[E(K, T_N)]$ |

- The counter value must be a nonce. That is, all $T_i$ across all messages must be unique.
- If a plaintext block is encrypted using a given counter value is known, then the output of the encryption block is known from the associated ciphertext. This output allows other plaintext blocks that are encrypted using the same counter value to be easily recovered from the associated ciphertext blocks.

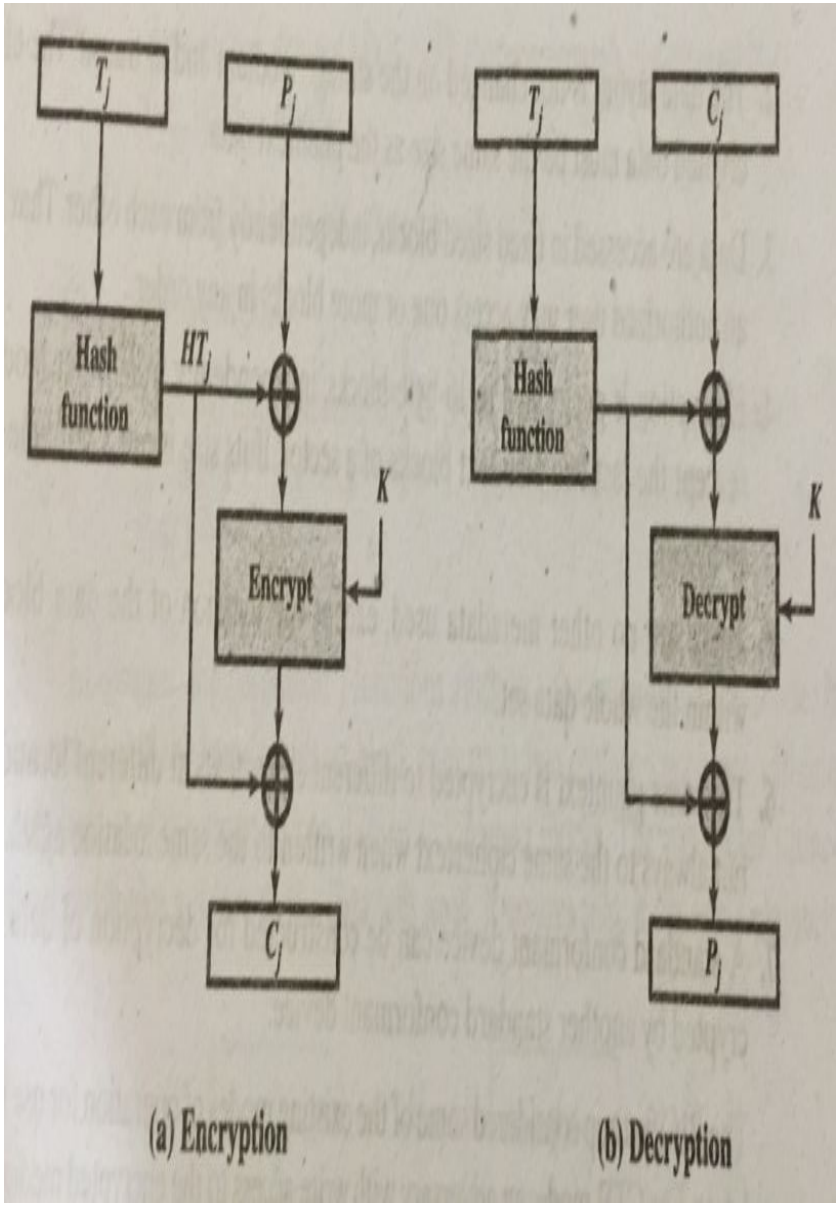
# Counter Mode

- Encryption and decryption can be performed in parallel. The throughput is only limited by the amount of parallelism that is achieved.
- For the scope of parallel execution, processor can support parallel features such as aggressive pipelining, multiple instruction dispatch per clock cycle, a large number of registers and SIMD instruction, etc can be effectively utilized.
- Preprocessing of the encryption algorithm are possible.
- Random access is possible as chaining mode is not used.
- Unlike ECB and CBC modes, CTR mode  requires only the implementation of the encryption algorithm.

# XTS-AES Mode

- In 2010, NIST approved additional block cipher mode of operation, XTS-AES.
- This mode is also an IEEE standard: IEEE Std 1619-2007.
- A method for encryption for data stored in sector-based devices.

# Tweakable Block Cipher




- Used as a basis of XTS-AES mode.
- The purpose of tweak is to introduce variability and it is not secret.
- ***C= H(T) XOR E(K, H(T) XOR P)***
- For decryption,

  ***H(T) XOR C = E(K, H(T) XOR P)***

  ***D[K, H(T) XOR C] = H(T) XOR P***

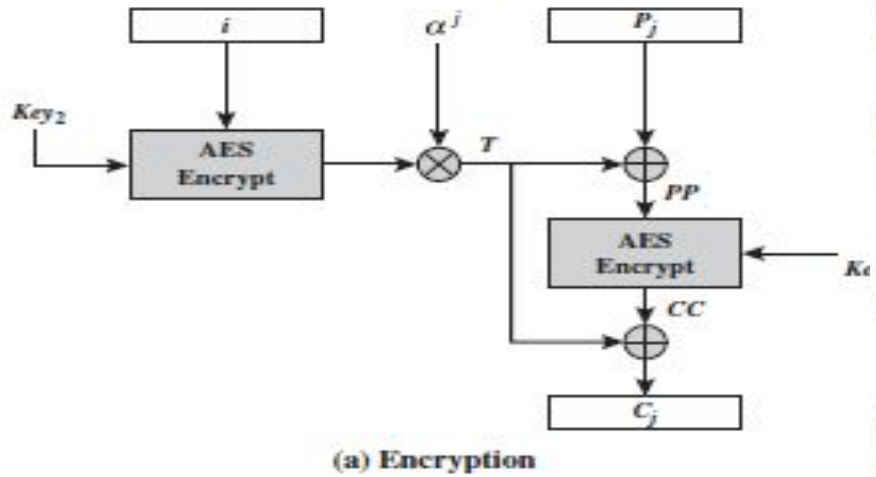
  ***H(T) XOR D[K, H(T) XOR C]=***

  ***H(T) XOR H(T) XOR P = P.***
- Tweak cipher uses ECB by removing the security issue of ECB.

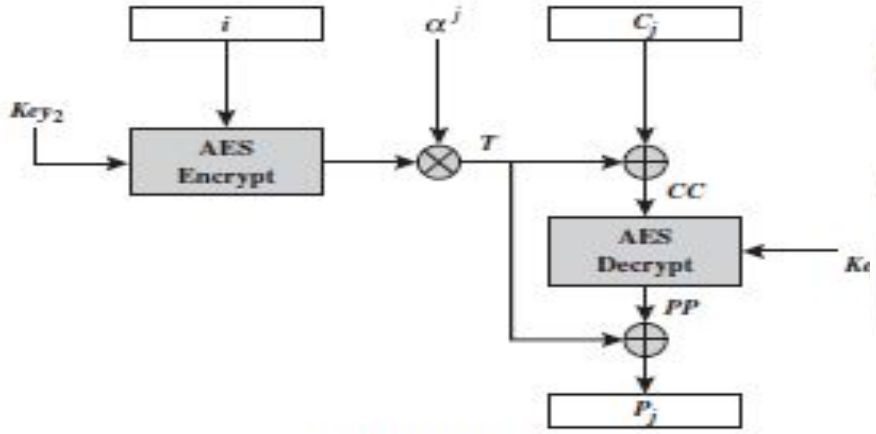
# Storage Encryption Requirements

1. The ciphertext is freely available for an attacker. Among the circumstances that lead to this situation:
   a. A group of users has authorized access to a database. Some of the records in the database are encrypted so that only specific users can successfully read/write them. Other users can retrieve an encrypted record but are unable to read it without the key.
   b. An unauthorized user manages to gain access to encrypted records.
   c. A data disk or laptop is stolen, giving the adversary access to the encrypted data.
2. The data layout is not changed on the storage medium and in transit. The encrypted data must be the same size as the plaintext data.
3. Data are accessed in fixed sized blocks, independently from each other. That is, an authorized user may access one or more blocks in any order.
4. Encryption is performed in 16-byte blocks, independently from other blocks (except the last two plaintext blocks of a sector, if its size is not a multiple of 16 bytes).
5. There are no other metadata used, except the location of the data blocks within the whole data set.
6. The same plaintext is encrypted to different ciphertexts at different locations, but always to the same ciphertext when written to the same location again.
7. A standard conformant device can be constructed for decryption of data encrypted by another standard conformant device.

# Operation on a Single Block




(a) Encryption

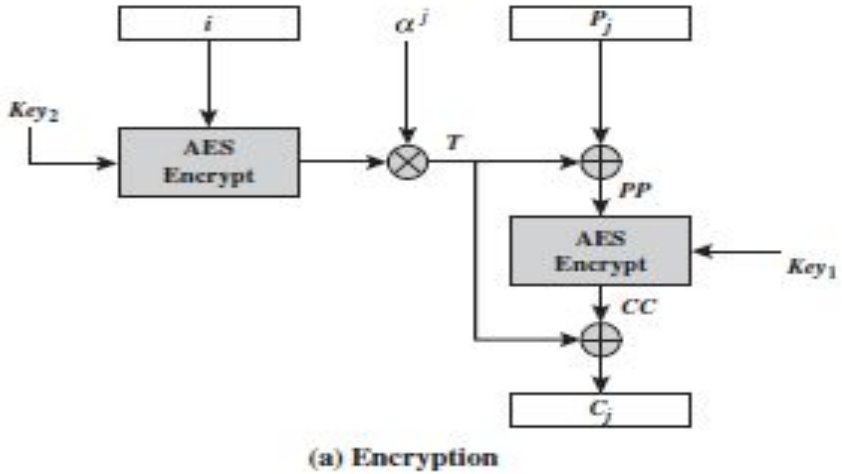



(b) Decryption

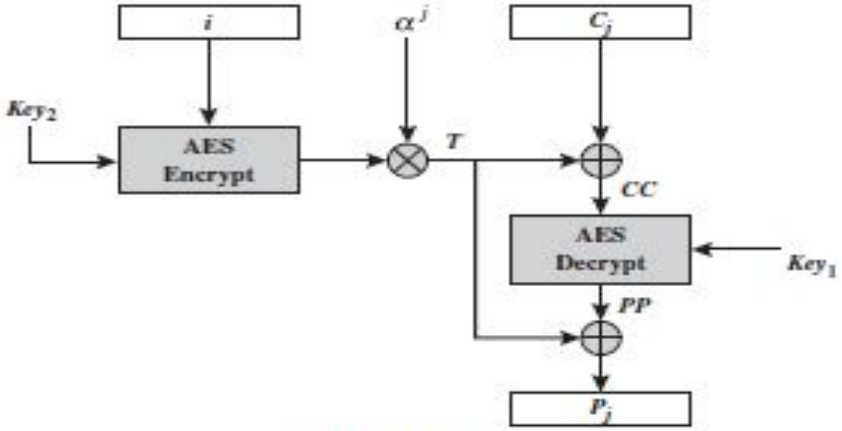
Figure 6.9 XTS-AES Operation on Single Block

| | |
|---|---|
| $Key$ | The 256 or 512 bit XTS-AES key; this is parsed as a concatenation of two fields of equal size called $Key_1$ and $Key_2$, such that $Key = Key_1 \| Key_2$. |
| $P_j$ | The $j$th block of plaintext. All blocks except possibly the final block have a length of 128 bits. A plaintext data unit, typically a disk sector, consists of a sequence of plaintext blocks $P_1, P_2, \ldots, P_m$. |
| $C_j$ | The $j$th block of ciphertext. All blocks except possibly the final block have a length of 128 bits. |
| $j$ | The sequential number of the 128-bit block inside the data unit. |
| $i$ | The value of the 128-bit tweak. Each data unit (sector) is assigned a tweak value that is a nonnegative integer. The tweak values are assigned consecutively, starting from an arbitrary nonnegative integer. |
| $\alpha$ | A primitive element of $GF(2^{128})$ that corresponds to polynomial $x$ (i.e., $0000\ldots010_2$). |
| $\alpha^j$ | $\alpha$ multiplied by itself $j$ times, in $GF(2^{128})$. |
| $\oplus$ | Bitwise XOR. |
| $\otimes$ | Modular multiplication of two polynomials with binary coefficients modulo $x^{128} + x^7 + x^2 + x + 1$. Thus, this is multiplication in $GF(2^{128})$. |

# Operation on a Single Block




(a) Encryption




(b) Decryption

Figure 6.9 XTS-AES Operation on Single Block

| | | |
|---|---|---|
| XTS-AES block operation | $T = E(K_2, i) \otimes \alpha^j$<br>$PP = P \oplus T$<br>$CC = E(K_1, PP)$<br>$C = CC \oplus T$ | $T = E(K_2, i) \otimes \alpha^j$<br>$CC = C \oplus T$<br>$PP = D(K_1, CC)$<br>$P = PP \oplus T$ |

- For encryption,

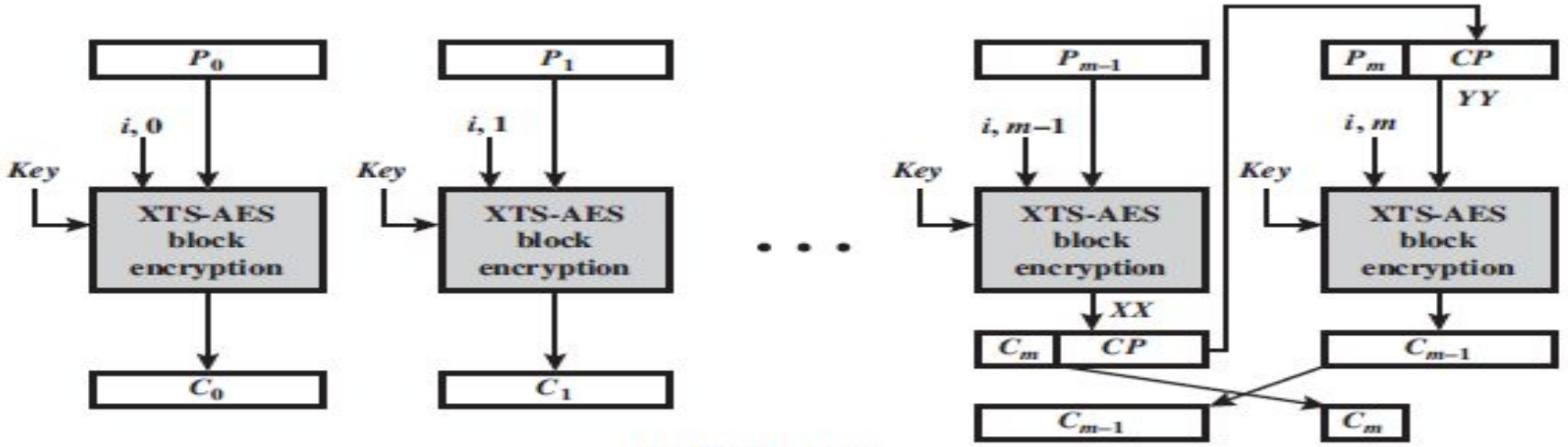
$$C = CC \oplus T = E(K_1, PP) \oplus T = E(K_1, P \oplus T) \oplus T$$

- For decryption,

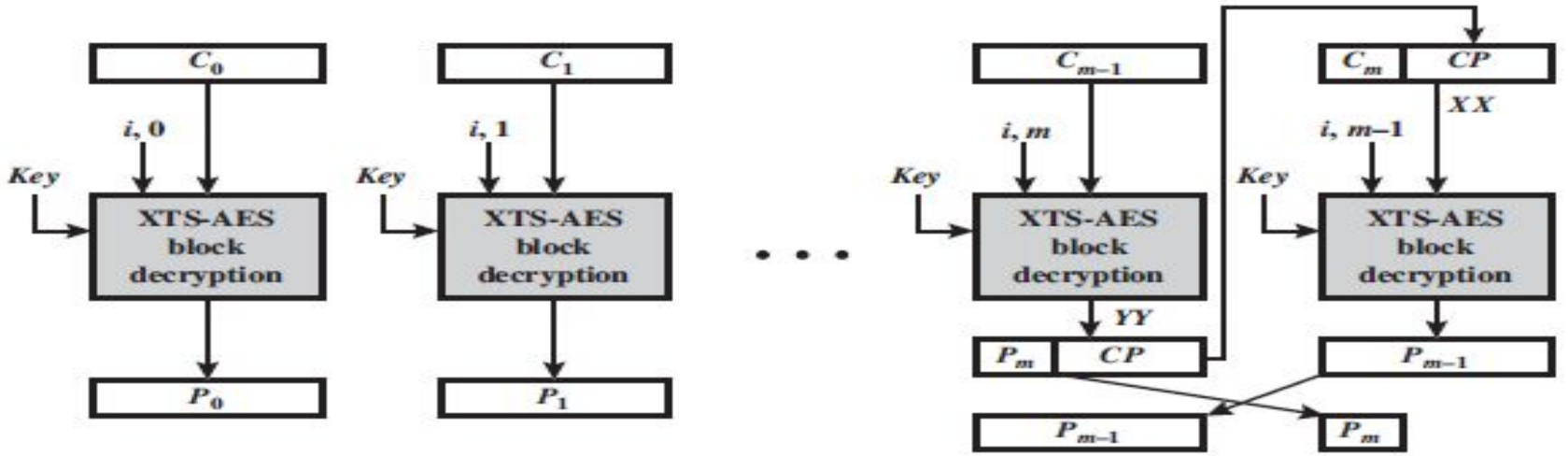
$$P = PP \oplus T = D(K_1, CC) \oplus T = D(K_1, C \oplus T) \oplus T$$

$$\begin{aligned} P &= D(K_1, C \oplus T) \oplus T \\ &= D(K_1, [E(K_1, P \oplus T) \oplus T] \oplus T) \oplus T \\ &= D(K_1, E(K_1, P \oplus T)) \oplus T \\ &= (P \oplus T) \oplus T = P \end{aligned}$$

# Operation on a Sector




(a) Encryption




(b) Decryption

# Operation on a Sector

| | |
|---|---|
| XTS-AES mode with null final block | $C_j = \text{XTS-AES-blockEnc}(K, P_j, i, j) \quad j = 0, \ldots, m-1$ |
| | $P_j = \text{XTS-AES-blockEnc}(K, C_j, i, j) \quad j = 0, \ldots, m-1$ |
| XTS-AES mode with final block containing $s$ bits | $C_j = \text{XTS-AES-blockEnc}(K, P_j, i, j) \quad j = 0, \ldots, m-2$<br>$XX = \text{XTS-AES-blockEnc}(K, P_{m-1}, i, m-1)$<br>$CP = \text{LSB}_{128-s}(XX)$<br>$YY = P_m \| CP$<br>$C_{m-1} = \text{XTS-AES-blockEnc}(K, YY, i, m)$<br>$C_m = \text{MSB}_s(XX)$ |
| | $P_j = \text{XTS-AES-blockDec}(K, C_j, i, j) \quad j = 0, \ldots, m-2$<br>$YY = \text{XTS-AES-blockDec}(K, C_{m-1}, i, m-1)$<br>$CP = \text{LSB}_{128-s}(YY)$<br>$XX = C_m \| \text{CP}$<br>$P_{m-1} = \text{XTS-AES-blockDec}(K, XX, i, m)$<br>$P_m = \text{MSB}_s(YY)$ |

- Suitable for parallel operation like the CTR mode.