Block Cipher Encryption Modes

Dept. of Computer Science & Engg.
IIT Kharagpur, India

Teacher: Dipanwita Roy Chowdhury

Multiple Encryption & DES

- DES is broken
  - theoretical attacks that can break it
  - 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

- Use 2 DES encrypts on each block
  - $C = E_{K2}(E_{K1}(P))$

- and have "meet-in-the-middle" attack
  - works whenever use a cipher twice
  - since $X = E_{K1}(P) = D_{K2}(C)$
  - attack by encrypting P with all keys and store
  - then decrypt C with keys and match X value
  - can show takes $O(2^{56})$ steps

Meet in the Middle Attack

- To improve the security of a block cipher, one might get the (naive) idea to simply use two independent keys to encrypt the data twice.
- $C = EK_2 [EK_1 [P]]$
- Naively, one might think that this would square the security of the double-encryption scheme.
- In fact, an exhaustive search of all possible combinations of keys would take 2^{2n} attempts (if each key K_1, K_2 is n bits long), compared to the 2^n attempts required for searching a single key.

Meet in the Middle Attack

- Assume the attacker knows a set of Plaintext (P) and Ciphertext (C). That is, C = EK2 [EK1 [P]] where E is the encryption function (cipher), and K1 and K2 are the two keys.
- The attacker can first compute EK(P) for all possible keys K and store the results in memory (in a lookup table).
- The ciphertext can be decrypted by computing DK(C) for each K.
- Any matches between these two resulting sets are likely to reveal the correct keys. (To speed up the comparison, the EK(P) set is stored in an in memory lookup table, then each DK(C) can be matched against the values in the lookup table to find the candidate keys.) •
- Once the matches are discovered, they can be verified with a second test set of Plaintext and Ciphertext.
- If the key-size is n, this attack uses only 2^{n+1} (for Double DES, $2^{56+1}=2^{57}$) encryptions/decryptions (and $O(2^n)$ memory space) in contrast to the naive attack, which needs 22n encryptions/decryptions (but only O(1) space).

Triple-DES

- Use three different keys
  – Encrypt: $C = EK_3 [DK_2 [EK_1 [P]]]$
  – Decrypt: $P = DK_1 [EK_2 [DK_3 [C]]]$
  Case 1: All three keys are identical, i.e. $K_1 = K_2 = K_3$
  Case 2: K_1 and K_2 are independent, and $K_3 = K_1$.
  Case 3: All three keys are independent
    - The key space is 56 x 3 = 168 bits
    - No known practical attack against it.
    - Many protocols/applications use 3DES (example PGP)
    - The electronic payment industry uses Triple DES and continues to develop and promulgate standards based upon it (Europay-Visa-Mastercard).

Triple-DES Security

- Case 1 is equivalent to DES, with only 56 key bits.

- Case 2 with $2 \times 56 = 112$ key bits provides less security than case 3. However, this option is stronger than double DES (with K 1 and K 2), because it protects against meet-in-the-middle attacks.

Note that this option is susceptible to certain chosen-plaintext or known-plaintext attacks, and thus it is designated by NIST to have only 80 bits of real security.

- Case 3: $K1 \neq K2 \neq K3$

– Encrypt: $C = E_{K3} [D_{K2} [E_{K1} [P]]]$

– Decrypt: $P = D_{K1} [E_{K2} [D_{K3} [C]]]$

Will the effective strength od case 3 be that of 56x3= 168 bits?

Modes of Operation

- block ciphers encrypt fixed size blocks
  - DES encrypts 64-bit blocks with 56-bit key
  - AES encrypts 128-bit with 128-bit key
- need some way to en/decrypt arbitrary amounts of data in practise
- NIST (FIPS 81) defines 4 possible modes
- subsequently 5 defined for AES & DES
- have **block** and **stream** modes

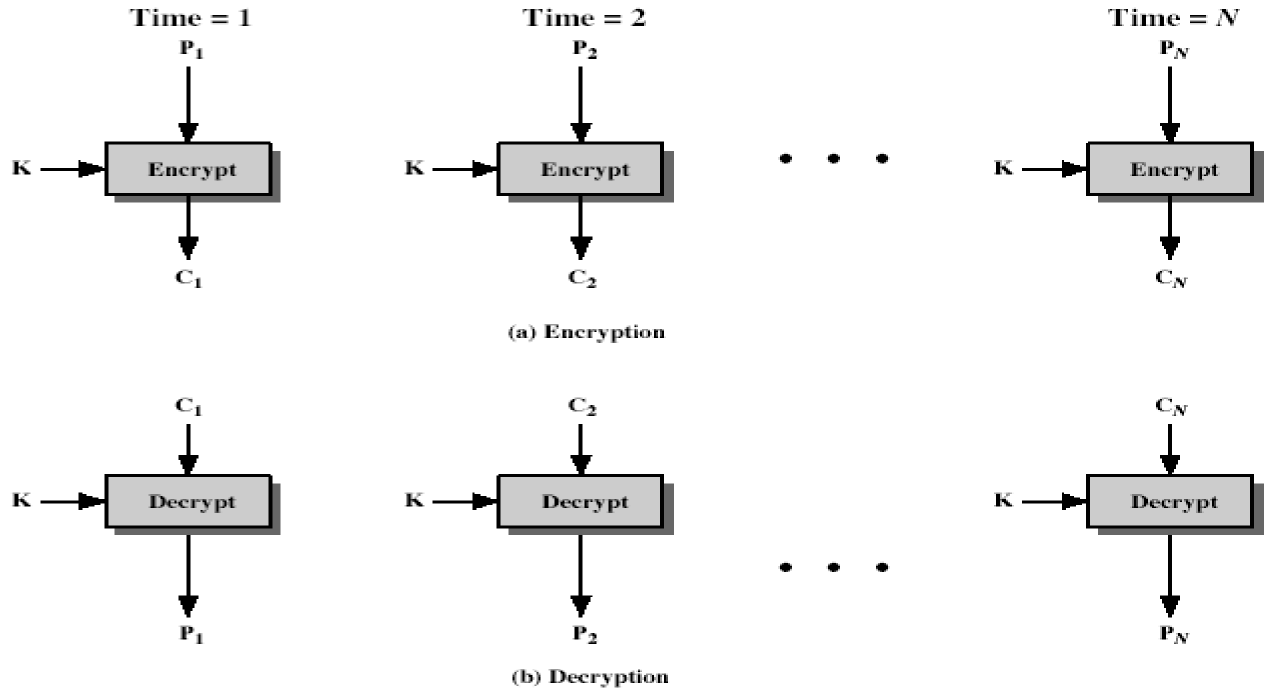
Electronic Codebook Book (ECB)

- Message is broken into independent blocks which are encrypted
- Each block is a value which is substituted, like a codebook, hence name
- Each block is encoded independently of the other blocks

  $C_i = E_{K1}(P_i)$

- Uses: secure transmission of single values

Electronic Codebook Book (ECB)



Stallings Book

Advantages and Limitations of ECB

- message repetitions may show in ciphertext
  - if aligned with message block
  - particularly with data such graphics
  - or with messages that change very little, which become a code-book analysis problem
- weakness is due to the encrypted message blocks being independent
- main use is sending a few blocks of data

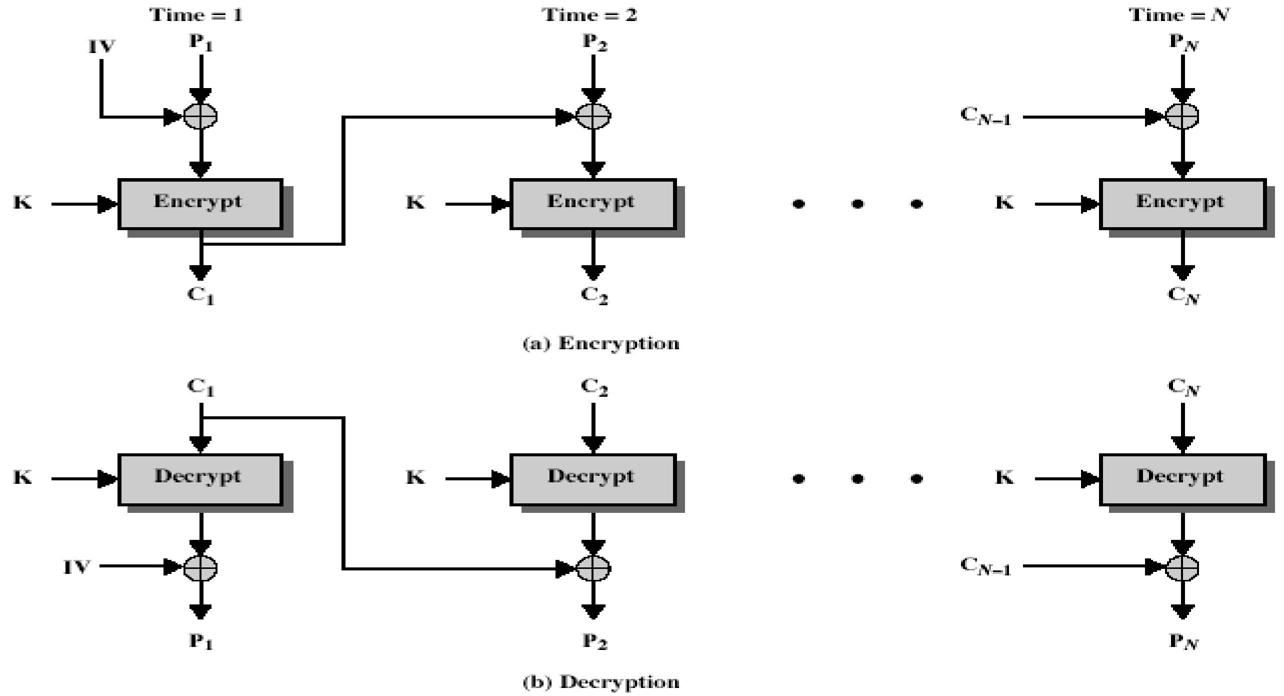
Cipher Block Chaining (CBC)

- message is broken into blocks
- linked together in encryption operation
- each previous cipher blocks is chained with current plaintext block, hence name
- use Initial Vector (IV) to start process

  $C_i = DES_{K1}(P_i \text{ XOR } C_{i-1})$

  $C_{-1} = IV$
- uses: bulk data encryption, authentication

Cipher Block Chaining (CBC)



Stallings Book

Message Padding

- at end of message must handle a possible last short block
  - which is not as large as block size of cipher
  - pad either with known non-data value (e.g. nulls)
  - or pad last block along with count of pad size
    - eg. [b1 b2 b3 0 0 0 0 5]
    - means have 3 data bytes, then 5 bytes pad + count
  - this may require an extra entire block over those in message
- there are other, more esoteric modes, which avoid the need for an extra block

Advantages and Limitations of CBC

- a ciphertext block depends on **all** blocks before it
- any change to a block affects all following ciphertext blocks
- need **Initialization Vector** (IV)
  - which must be known to sender & receiver
  - if sent in clear, attacker can change bits of first block, and change IV to compensate
  - hence IV must either be a fixed value (as in EFTPOS) or must be sent encrypted in ECB mode before rest of message

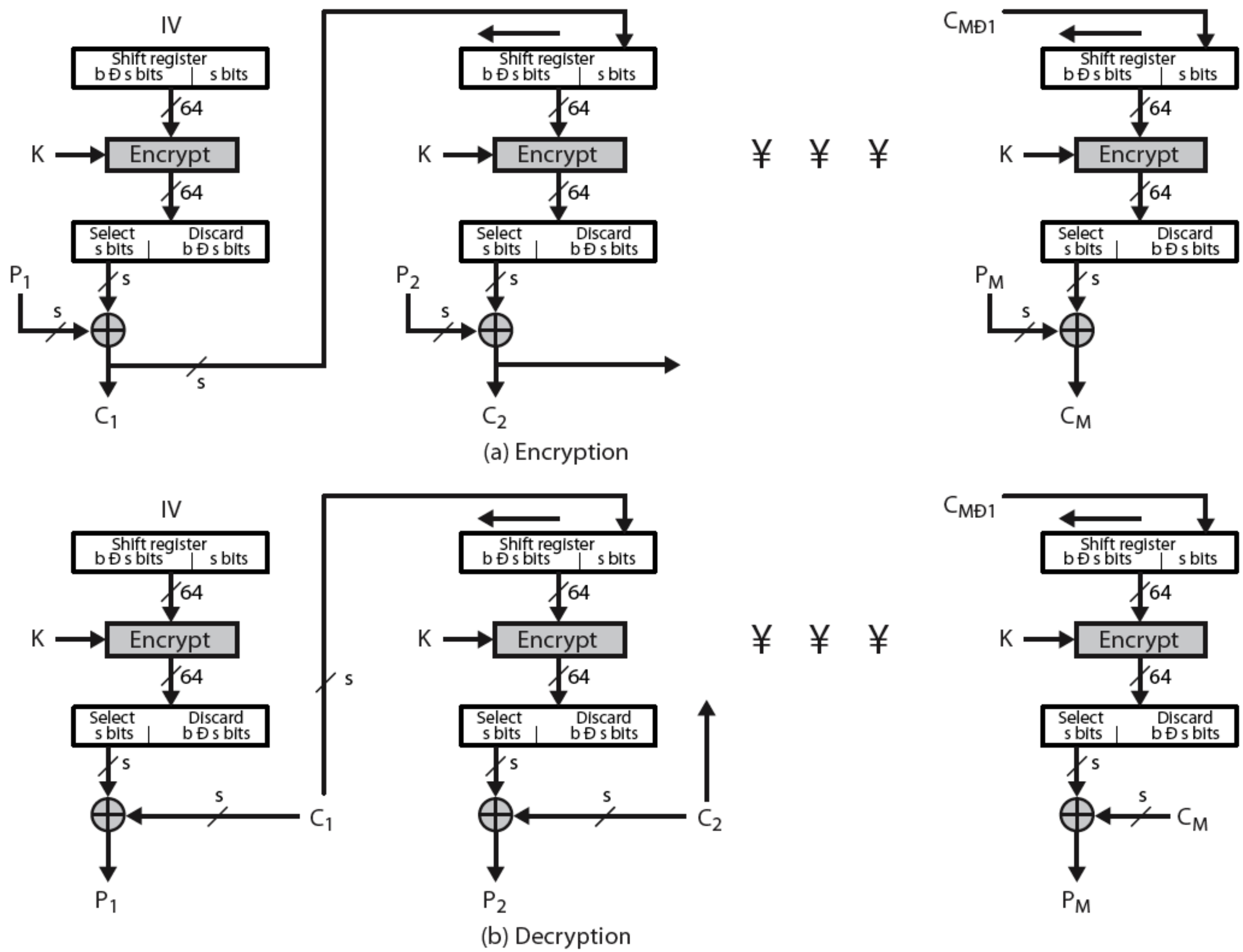
Cipher FeedBack (CFB)

- message is treated as a stream of bits
- added to the output of the block cipher
- result is feed back for next stage (hence name)
- standard allows any number of bit (1,8, 64 or 128 etc) to be feed back
  - denoted CFB-1, CFB-8, CFB-64, CFB-128 etc
- most efficient to use all bits in block (64 or 128)

  $C_i = P_i \text{ XOR } DES_{K1}(C_{i-1})$

  $C_{-1} = IV$
- uses: stream data encryption, authentication

Cipher FeedBack (CFB)



Stallings Book

Advantages and Limitations of CFB

- appropriate when data arrives in bits/bytes
- most common stream mode
- limitation is needed to stall while do block encryption after every n-bits
- note that the block cipher is used in **encryption** mode at **both** ends
- errors propogate for several blocks after the error

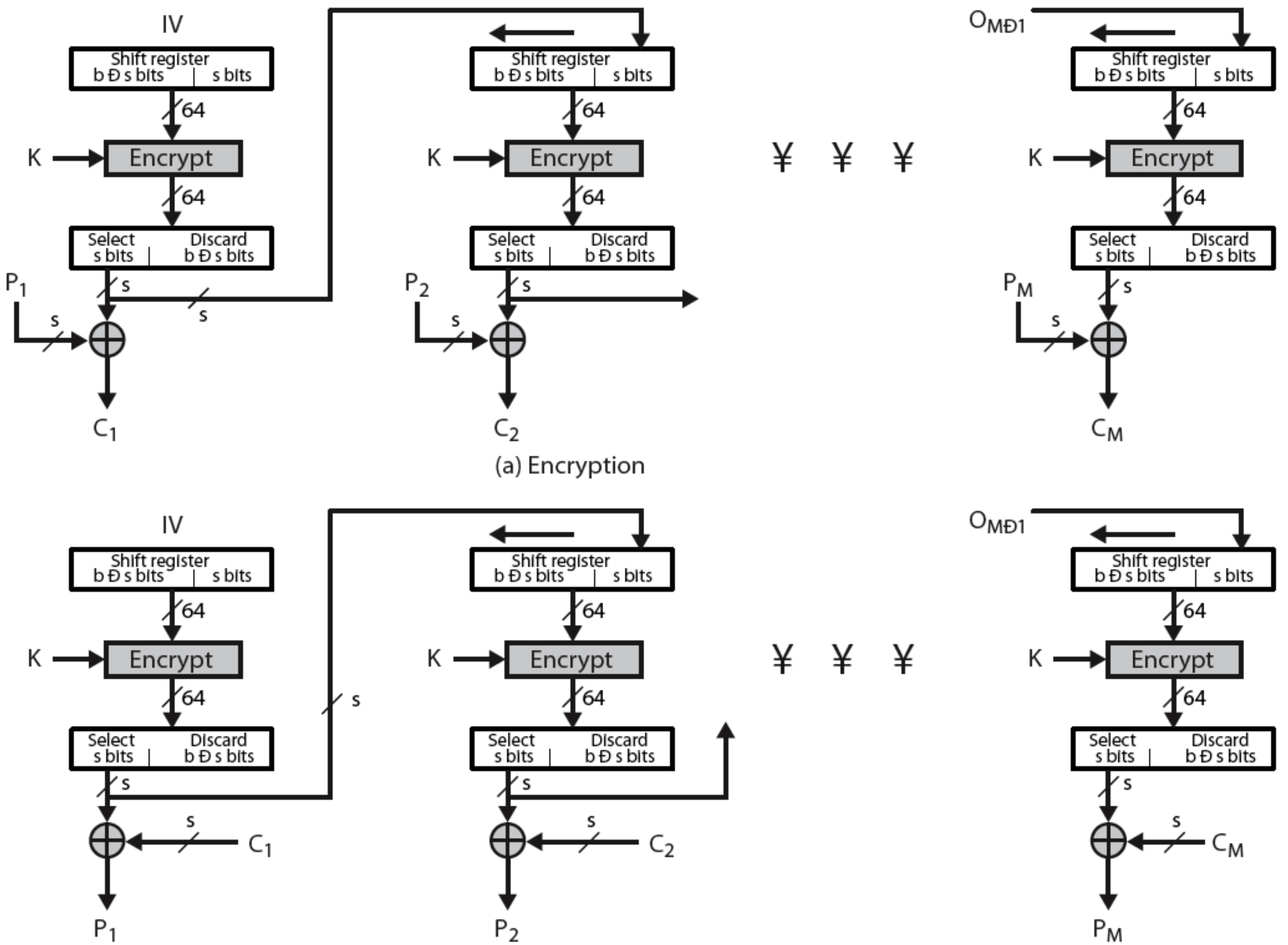
Output FeedBack (OFB)

- message is treated as a stream of bits
- output of cipher is added to message
- output is then feed back (hence name)
- feedback is independent of message
- can be computed in advance

$$C_i = P_i \text{ XOR } O_i$$
$$O_i = DES_{K1}(O_{i-1})$$
$$O_{-1} = IV$$

- uses: stream encryption on noisy channels

Output FeedBack (OFB)



(a) Encryption

(b) Decryption

Advantages and Limitations of OFB

- bit errors do not propagate
- more vulnerable to message stream modification
- a variation of a Vernam cipher
  - hence must **never** reuse the same sequence (key+IV)
- sender & receiver must remain in syncronized
- originally specified with m-bit feedback
- subsequent research has shown that only **full block feedback** (i.e. CFB-64 or CFB-128) should ever be used

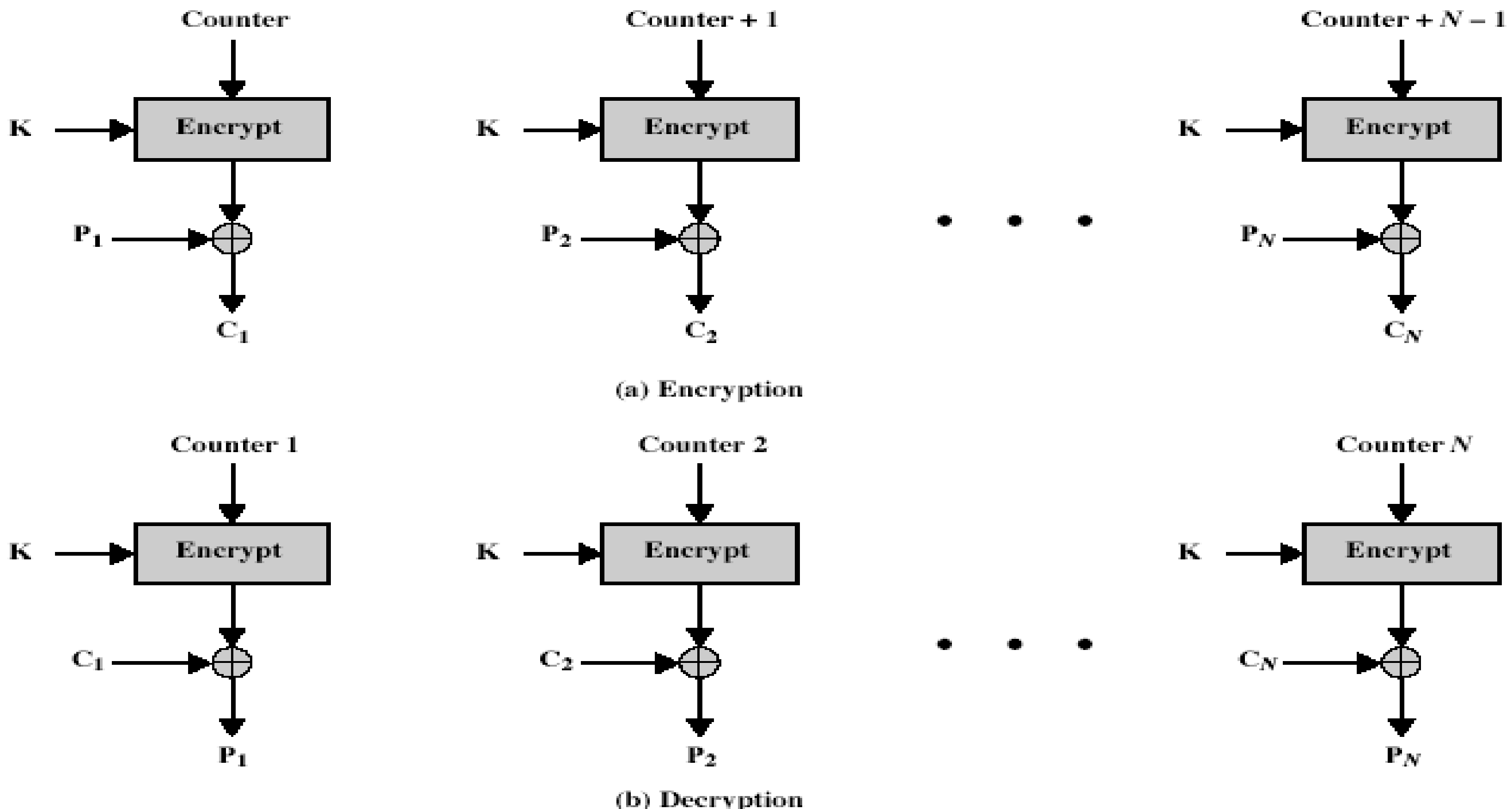
Counter (CTR) Mode

- a "new" mode, though proposed early on
- similar to OFB but encrypts counter value rather than any feedback value
- must have a different key & counter value for every plaintext block (never reused)

  $C_i = P_i \text{ XOR } O_i$

  $O_i = DES_{K1}(i)$
- uses: high-speed network encryptions

Counter (CTR)



Stallings Book

Advantages and Limitations of CTR

- Software and hardware efficiency: different blocks can be encrypted in parallel.
- Preprocessing: the encryption part can be done offline and when the message is known, just do the XOR.
- Random access: decryption of a block can be done in random order, very useful for hard-disk encryption.
- Messages of arbitrary length: ciphertext is the same length with the plaintext (i.e., no IV).