Information Systems Security

Dr. Ayman Abdel-Hamid

College of Computing and Information Technology

Arab Academy for Science & Technology and Maritime Transport

Chapters 3 and 6

Block Ciphers and DES

Outline

- Block Cipher Principles
- DES
- Block Cipher Modes of Operation
- Triple DES

Modern Block Ciphers

- will now look at modern block ciphers
- one of the most widely used types of cryptographic algorithms
- provide secrecy and/or authentication services
- in particular will introduce DES (Data Encryption Standard)

Block versus Stream Ciphers

- block ciphers process messages in blocks, each of which is then en/decrypted
  - (a block of plaintext is treated as a whole (typically 64 or 128 bits) and used to produce a ciphertext block of equal length)
- stream ciphers process messages a bit or byte at a time when en/decrypting
- many current ciphers are block ciphers

Block Cipher Principles

- most symmetric block ciphers are based on a **Feistel Cipher Structure**
- needed since must be able to **decrypt** ciphertext to recover messages efficiently
- block ciphers look like an extremely large substitution
- would need table of 2^{64} entries for a 64-bit block (for a *n*-bit block, 2^n different plaintext blocks exist, and $2^n!$ different transformations → reversible mappings)
- Arbitrary reversible substitution cipher for large *n* is not practical
- *Feistel* suggested to approximate a simple substitution cipher with a product cipher instead

Claude Shannon and Substitution-Permutation Ciphers

- in 1949 Claude Shannon introduced idea of substitution-permutation (S-P) networks
  - modern substitution-transposition product cipher
- these form the basis of modern block ciphers
- S-P networks are based on the two primitive cryptographic operations we have seen before:
  - *substitution* (S-box)
  - *permutation* (P-box)
- provide *confusion* and *diffusion* of message

Confusion and Diffusion

- cipher needs to completely obscure statistical properties of original message
- more practically Shannon suggested combining elements to obtain:
  - **diffusion** – dissipates statistical structure of plaintext over bulk of ciphertext
    - Example: average *k* successive letters
  - **confusion** – makes relationship between statistics of ciphertext and value of key as complex as possible

Feistel Cipher Structure 1/2

- *Horst Feistel* devised the **Feistel cipher**
  - based on concept of invertible product cipher
- partitions input block into two halves
  - process through multiple rounds which
  - perform a substitution on left data half
  - based on *round function* of right half & subkey
  - then have permutation swapping halves
- implements Shannon's substitution-permutation network concept

Feistel Cipher Structure 2/2

$L_i = R_{i-1}$

$R_i = L_{i-1}\ XOR\ \mathrm{F}(R_{i-1}, K_i)$

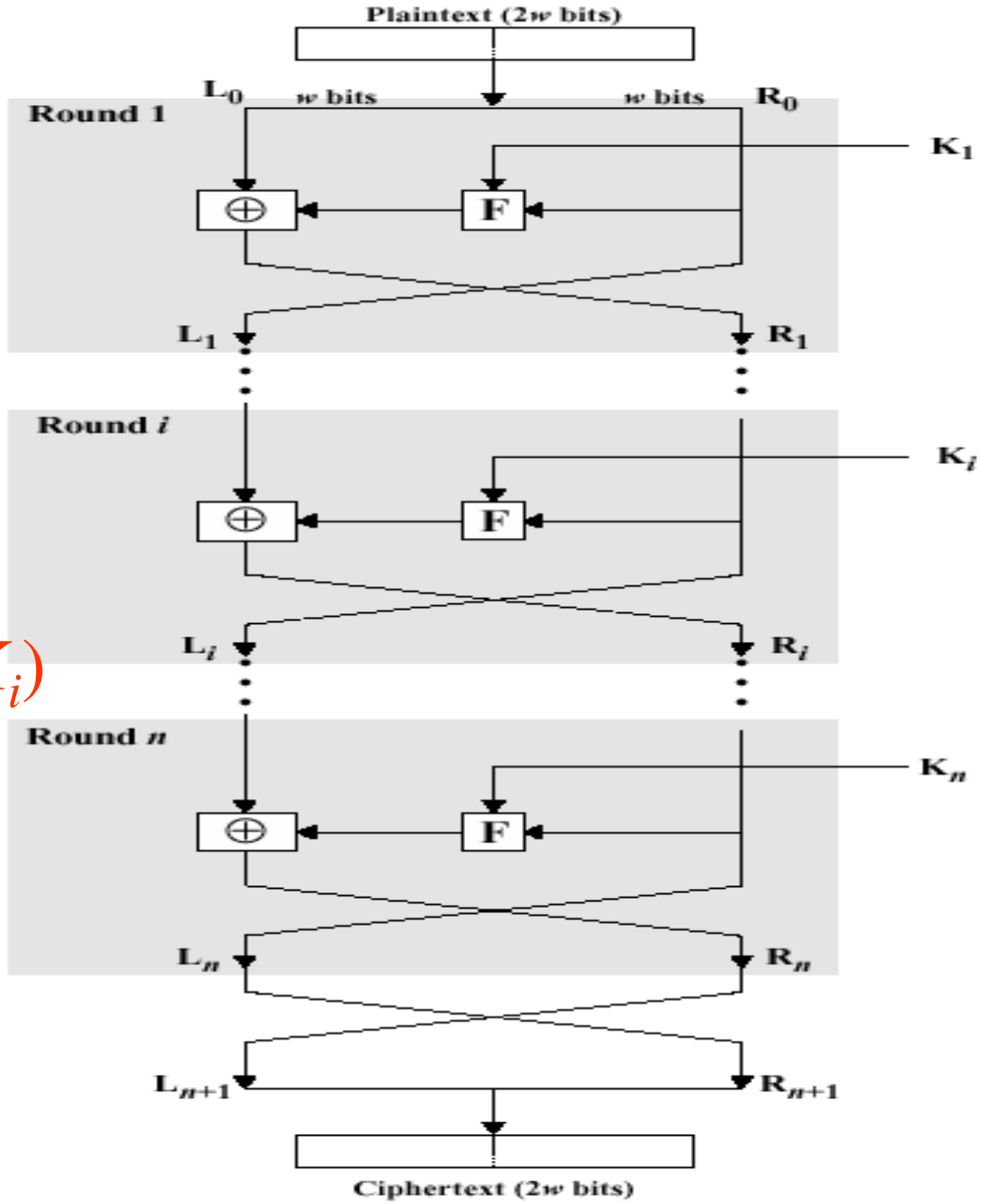
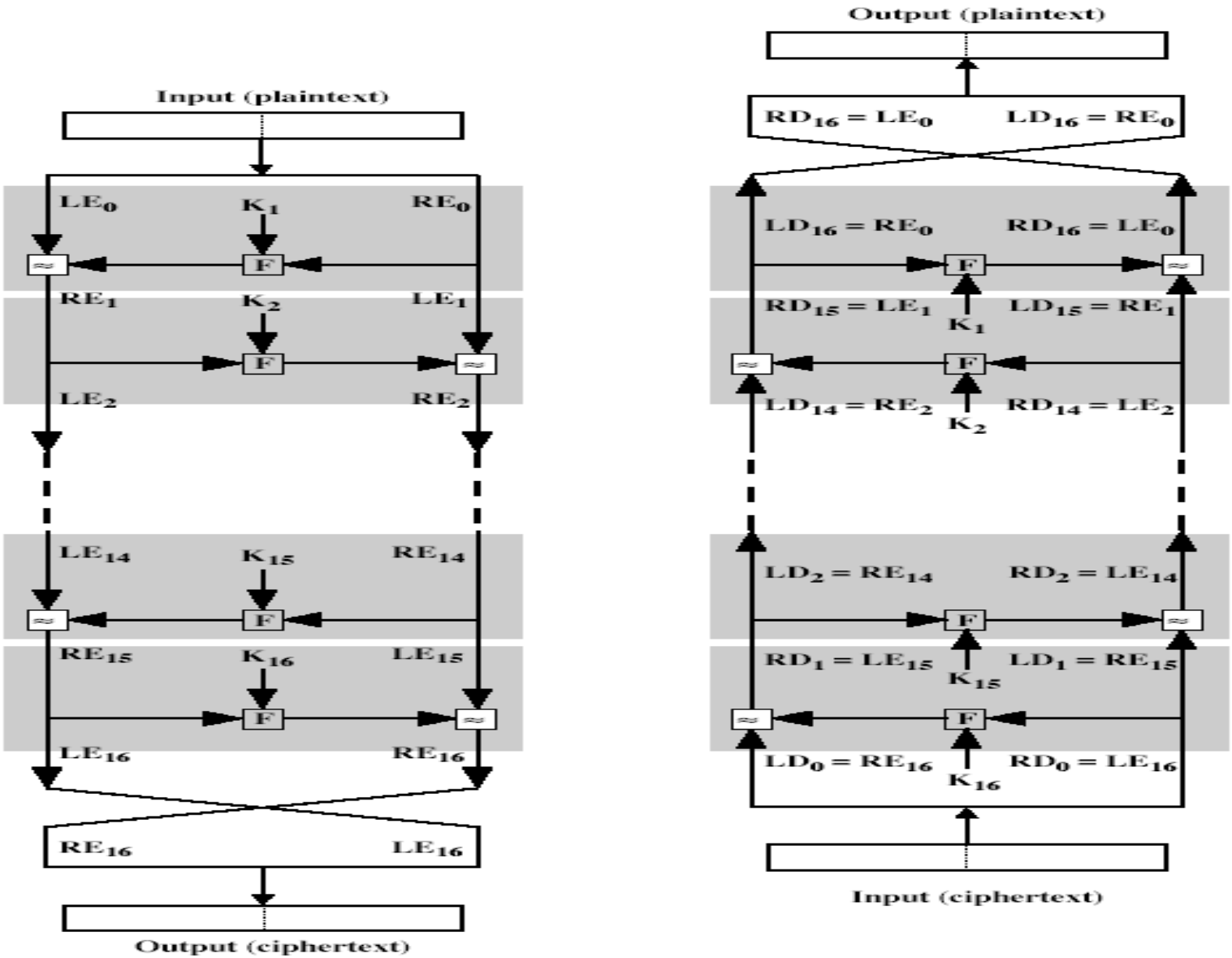


Figure 3.5 Classical Feistel Network

Feistel Cipher Design Principles

- **block size**
  - increasing size improves security, but slows cipher
- **key size**
  - increasing size improves security, makes exhaustive key searching harder, but may slow cipher
- **number of rounds**
  - increasing number improves security, but slows cipher (typical 16)
- **subkey generation**
  - greater complexity can make analysis harder, but slows cipher
- **round function**
  - greater complexity can make analysis harder, but slows cipher
- **fast software en/decryption & ease of analysis**
  - are more recent concerns for practical use and testing

Feistel Cipher Decryption



Data Encryption Standard (DES)

- most widely used block cipher in world
- adopted in 1977 by NBS (National Bureau of Standards) (now NIST: National Institute of Standards and Technology)
  - as FIPS (Federal Information Processing Standard) PUB 46
- encrypts 64-bit data using 56-bit key
- has widespread use
- has considerable controversy over its security

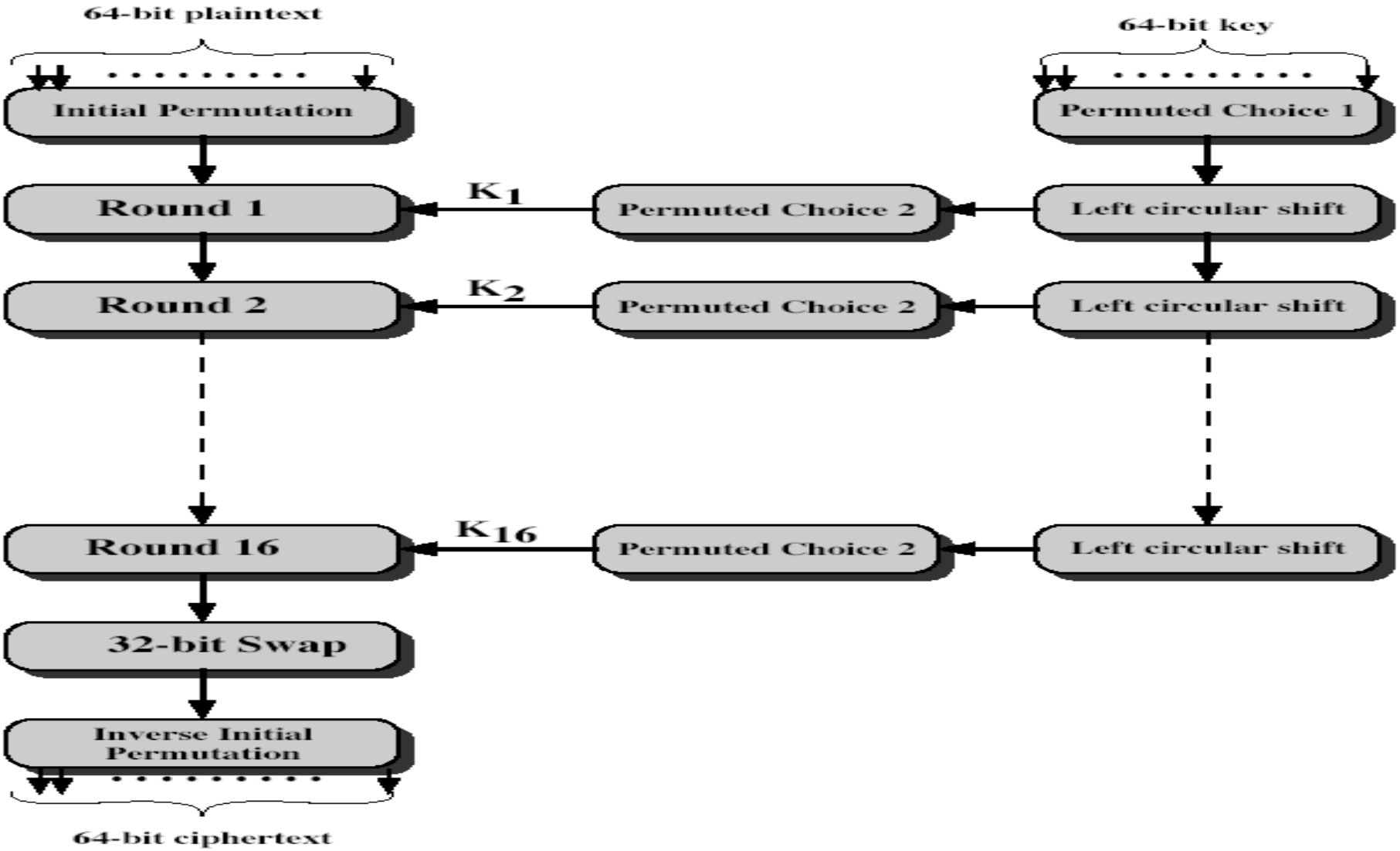
DES History

- IBM developed Lucifer cipher (1971)
  - by team led by Feistel
  - used 64-bit data blocks with 128-bit key
- then redeveloped as a commercial cipher with input from NSA and others
- in 1973, NBS issued request for proposals for a national cipher standard
- IBM submitted their revised Lucifer which was eventually accepted as the DES (1977)

DES Design Controversy

- although DES standard is public
- was considerable controversy over design
  - in choice of 56-bit key (versus Lucifer 128-bit)
  - and because design criteria were classified
- subsequent events and public analysis show in fact design was appropriate
- DES has become widely used, especially in financial applications

DES Encryption



Initial Permutation IP

- first step of the data computation
- IP reorders the input data bits
- see text Table 3.2

Table 3.2 Permutation Tables for DES

(a) Initial Permutation (IP)

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

(b) Inverse Initial Permutation (IP^{-1})

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

(c) Expansion Permutation (E)

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

(d) Permutation Function (P)

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

DES Round Structure 1/3

- uses two 32-bit L & R halves
- as for any Feistel cipher can describe as:

$$L_i = R_{i-1}$$

$$R_i = L_{i-1}\ XOR\ \mathrm{F}(R_{i-1}, K_i)$$

- takes 32-bit R half and 48-bit subkey and:
  - expands R to 48-bits using perm E
  - adds to subkey
  - passes through 8 S-boxes to get 32-bit result
  - finally permutes this using 32-bit perm P

DES Round Structure 2/3

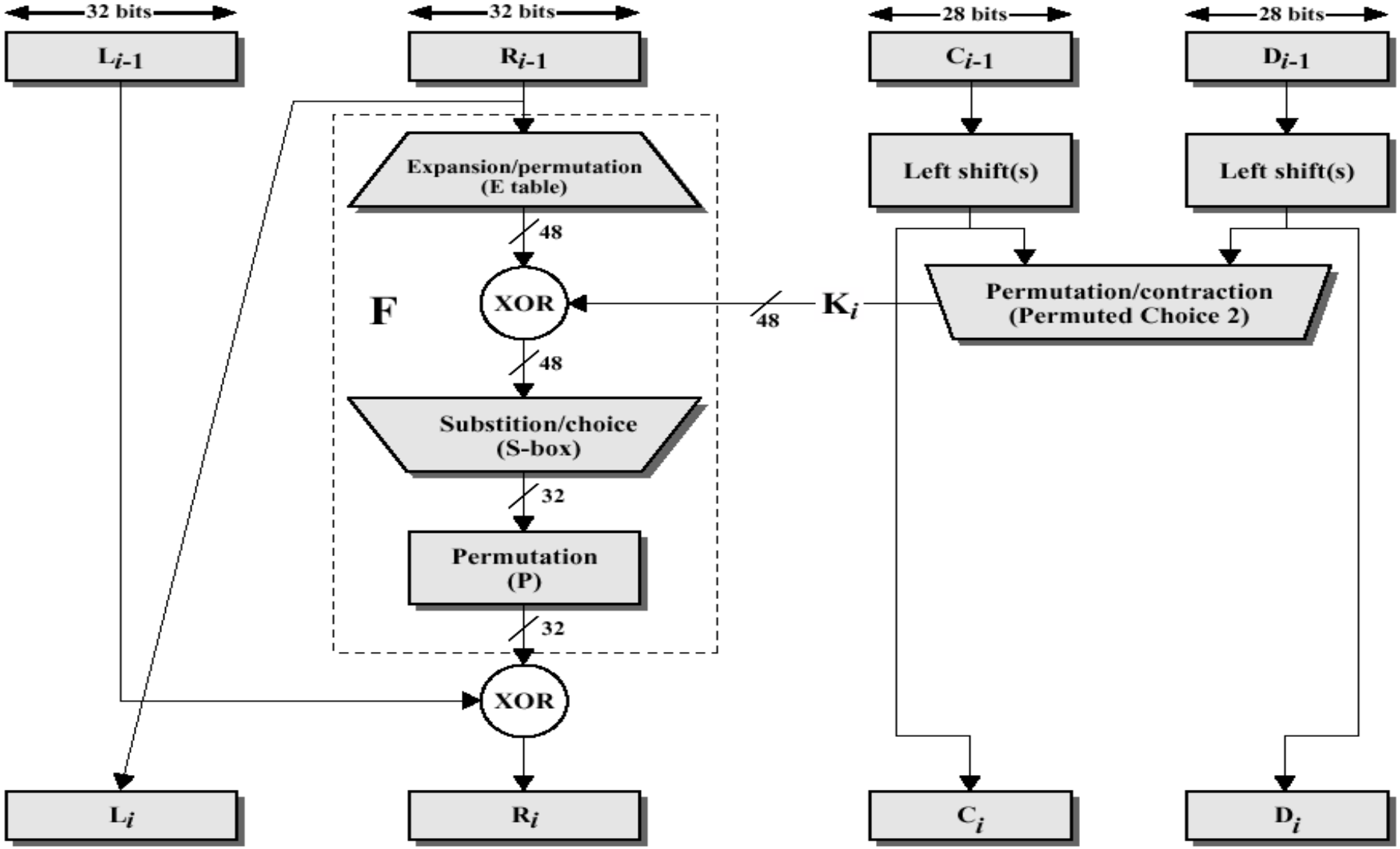
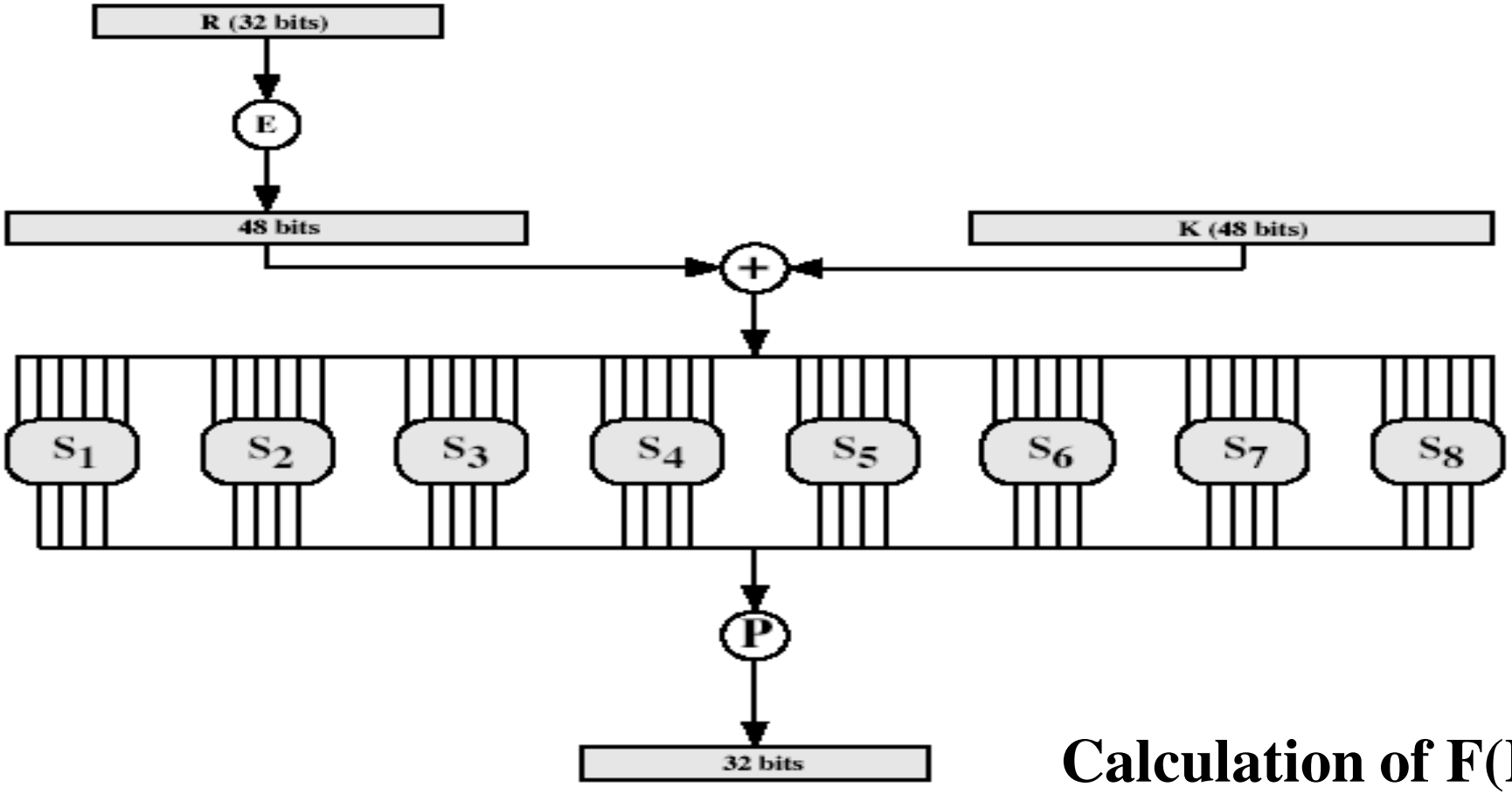


Figure 3.8 Single Round of DES Algorithm

DES Round Structure 3/3



Calculation of F(R,K)

Substitution Boxes S

- have eight S-boxes which map 6 to 4 bits
- each S-box is actually 4 little 4 bit boxes
  - outer bits 1 & 6 (**row** bits) select one row
  - inner bits 2-5 (**col** bits) are substituted
  - result is 8 lots of 4 bits, or 32 bits (from S-Boxes table 3.3)
- row selection depends on both data & key
  - feature known as autoclaving (autokeying)

DES Key Schedule

- forms subkeys used in each round
- 64 bit-key input with every 8^{th} bit ignored
- consists of:
  - initial permutation of the key (PC1) which selects 56-bits in two 28-bit halves
  - 16 stages consisting of:
    - selecting 24-bits from each half
    - permuting them by PC2 for use in function F,
    - rotating **each half** separately either 1 or 2 places depending on the **key rotation schedule** K

DES Decryption

- decrypt must unwind steps of data computation
- with Feistel design, do encryption steps again
- using subkeys in reverse order

Avalanche Effect

- key desirable property of encryption algorithm → *a small change in either the plaintext or the key should produce a significant change in ciphertext*
- making attempts to "home-in" by guessing keys impossible
- DES exhibits strong avalanche

Strength of DES – Key Size

- 56-bit keys have $2^{56} \approx 7.2 \times 10^{16}$ possible keys
- brute force search looks hard (more than a 1000 years)
- recent advances have shown is possible
  - In 1977, Diffie and Hellman → build a machine worth $20 millions that would bring down search time to about 10 hours
  - In 1998, EFF (Electronic Frontier Foundation) built a DES Cracker machine for less than $250,000 (published detailed description of machine). Attack took < 3 days
- still must be able to recognize plaintext
- now considering alternatives to DES

Strength of DES – Timing Attacks

- attacks actual implementation of cipher
- use knowledge of consequences of implementation to derive knowledge of some/all subkey bits
- specifically use fact that calculations can take varying times depending on the value of the inputs to it

Block Cipher Modes of Operation 1/2

- block ciphers encrypt fixed size blocks
- e.g., DES encrypts 64-bit blocks, with 56-bit key
- need way to use in practice, given usually have arbitrary amount of information to encrypt
- four were defined for DES in ANSI standard **ANSI X3.106-1983 Modes of Use**
- subsequently now have 5 for DES and AES
- have **block** and **stream** modes

Block Cipher Modes of Operation 2/2

Table 3.6 Block Cipher Modes of Operation

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
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
Output Feedback (OFB)	Similar to CFB, 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

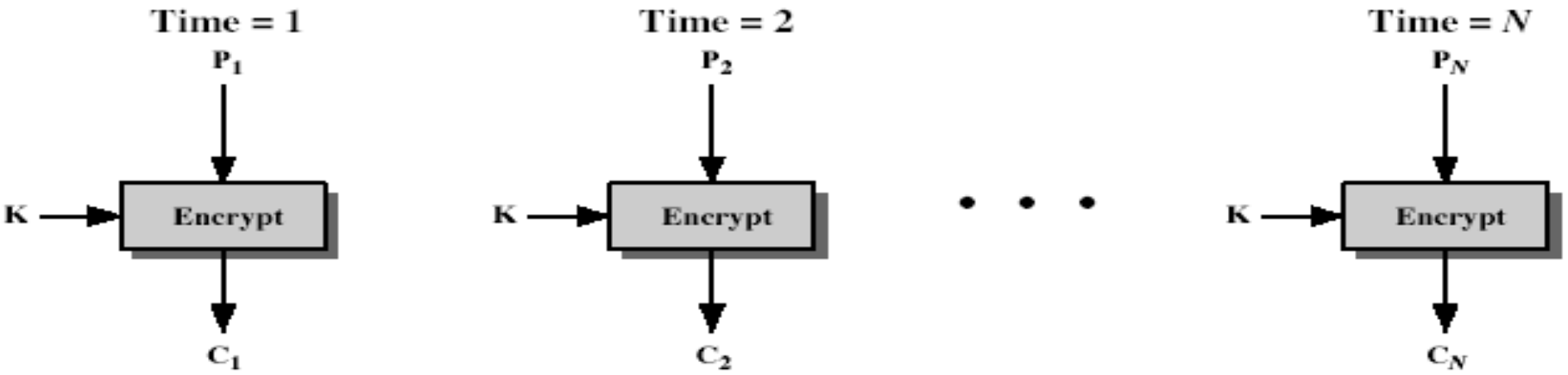
Electronic Codebook Mode (ECB) 1/2

- message is broken into independent blocks which are encrypted (pad last block if necessary)
- each block is a value which is substituted, like a *codebook*, hence name
- each block is encoded independently of the other blocks

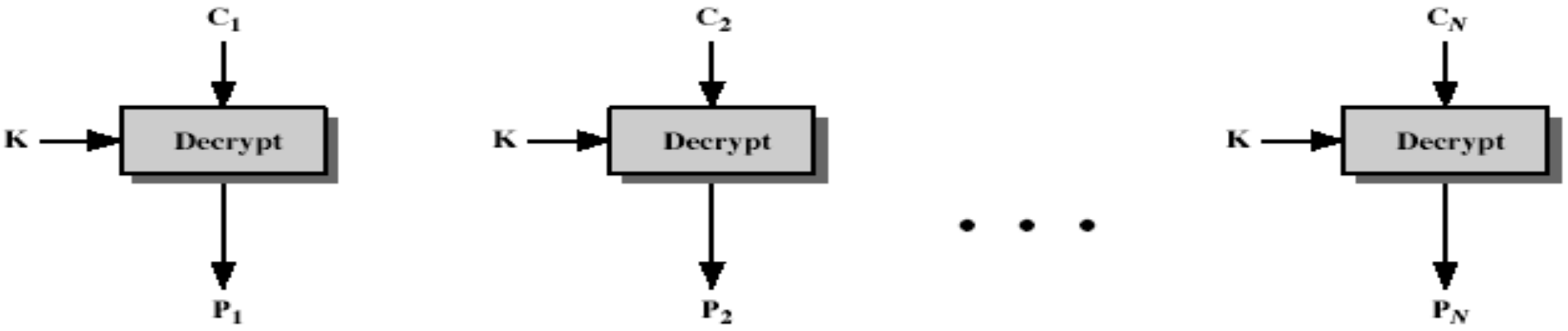
$$C_i = DES_{K1}(P_i)$$

- uses: secure transmission of single values

Electronic Codebook Mode (ECB) 2/2



(a) Encryption



(b) Decryption

Advantages and Limitations of ECB

- repetitions in message 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 due to encrypted message blocks being independent
- main use is sending a few blocks of data

Cipher Block Chaining (CBC) 1/2

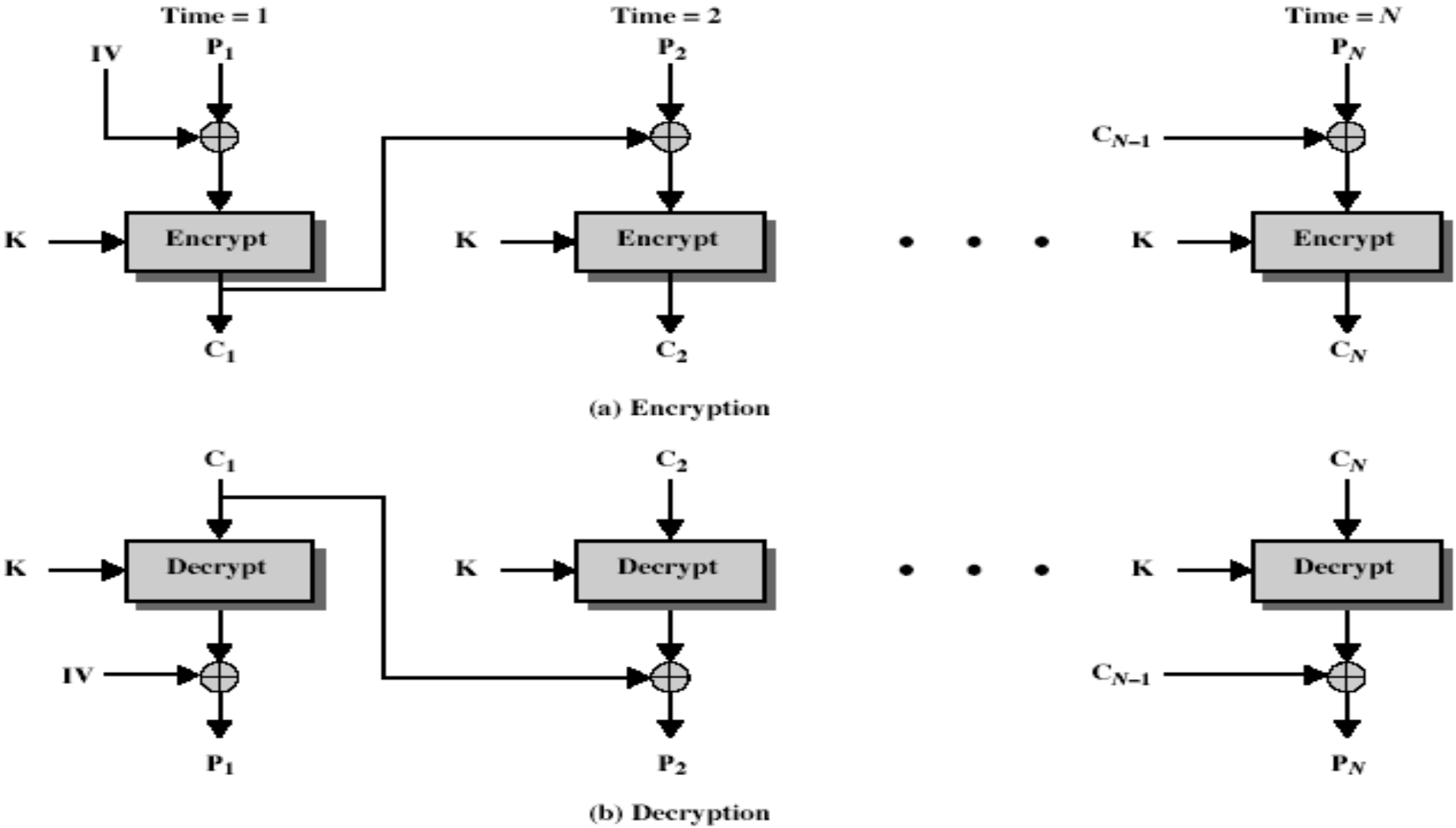
- message is broken into blocks
- but these are linked together in the encryption operation
- each previous cipher block is chained with current plaintext block, hence name
- use *Initialization Vector* (IV) to start process

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

$$C_{-1} = IV$$

- uses: bulk data encryption, authentication

Cipher Block Chaining (CBC) 2/2



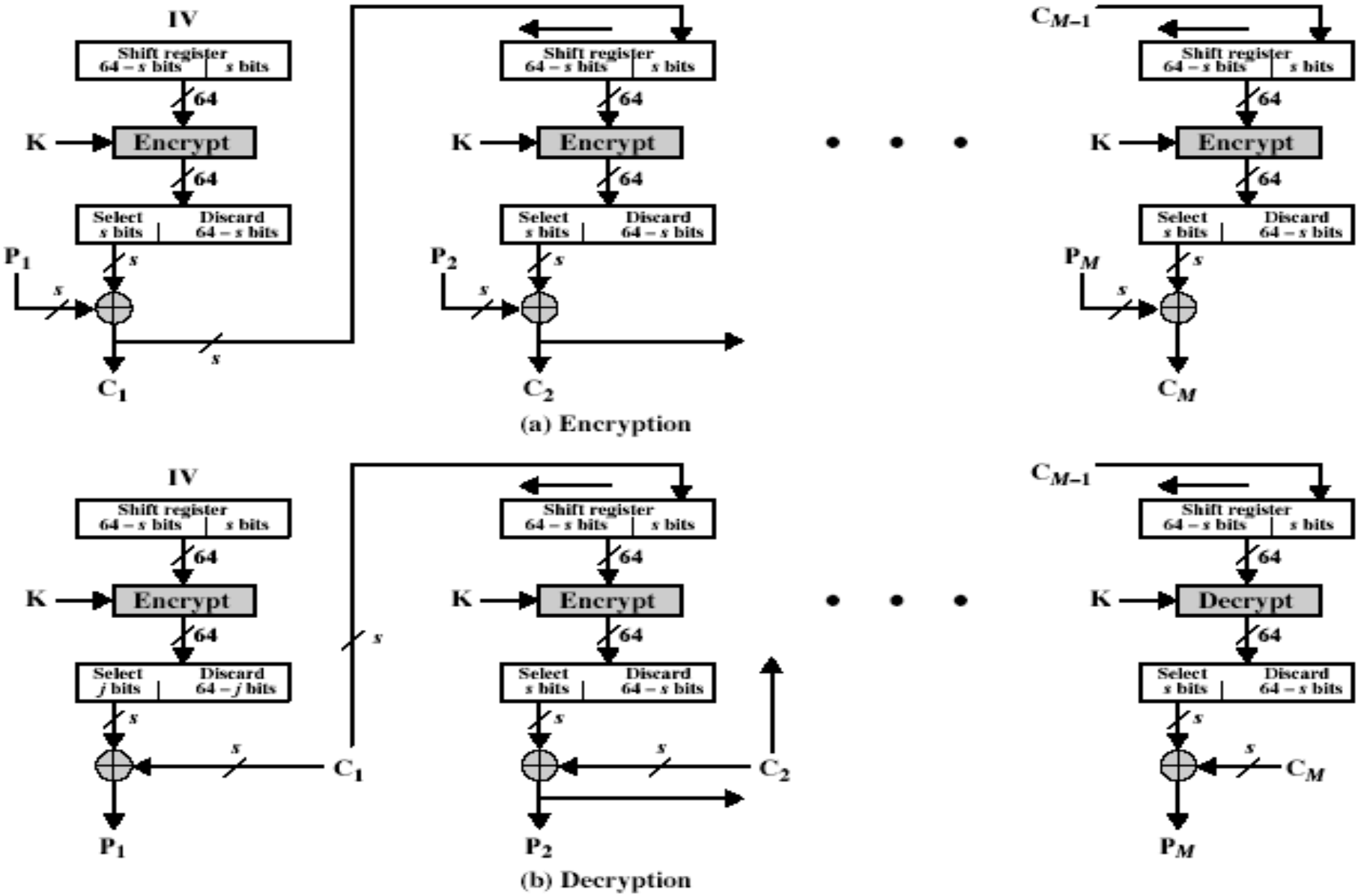
Advantages and Limitations of CBC

- each ciphertext block depends on **all** message blocks
- thus a change in the message affects all ciphertext blocks after the change as well as the original block
- need **Initialization Vector** (IV) known to sender & receiver
  - however if IV is sent in the clear, an attacker can change bits of the first block, and change IV to compensate
  - can be sent encrypted in ECB mode before rest of message

Cipher Feedback Mode (CFB) 1/2

- message is treated as a *stream of bits* (eliminate the need to pad the message to be an integral number of blocks)
- Input to encryption function is a 64-bit shift register (initially set to IV)
- Most significant *s* (common is 8. *s* is the unit of transmission) bits of encryption function are XORed with first segment of plaintext to produce first segment of ciphertext C_1 (transmitted)
- Contents of shift register are shifted left by *s* bits and C_1 placed in rightmost *s* bits
- uses: stream data encryption, authentication

Cipher Feedback Mode (CFB) 2/2



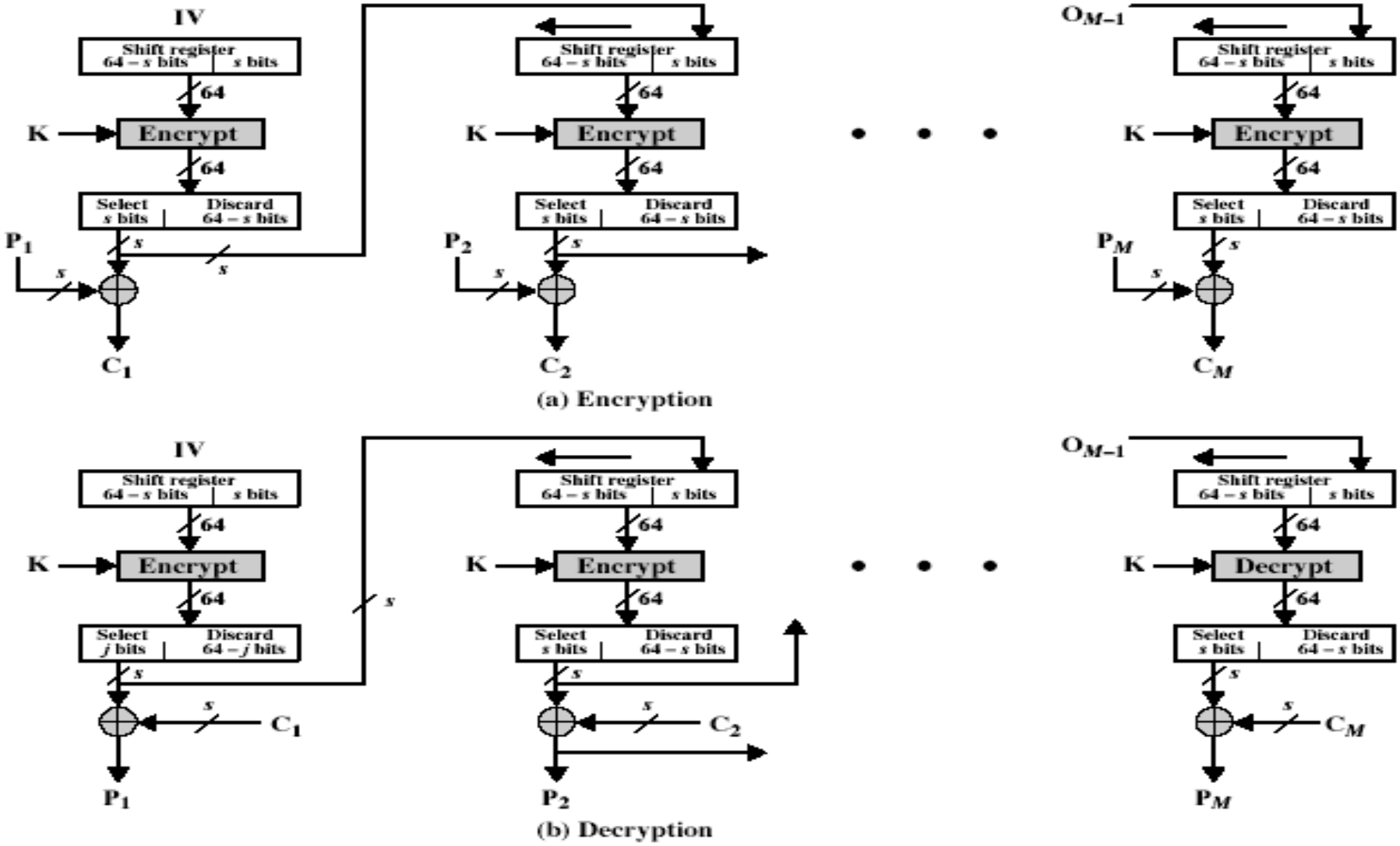
Advantages and Limitations of CFB

- appropriate when data arrives in bits/bytes
- most common stream mode
- limitation is *need to stall while do block encryption after every n-bits*
- note that the block cipher is used in **encryption** mode at **both** ends
- Bit errors in transmission might propagate

Output Feedback Mode (OFB) 1/2

- 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
- uses: stream encryption over noisy channels (bit errors in transmission do not propagate)

Output Feedback Mode (OFB) 2/2



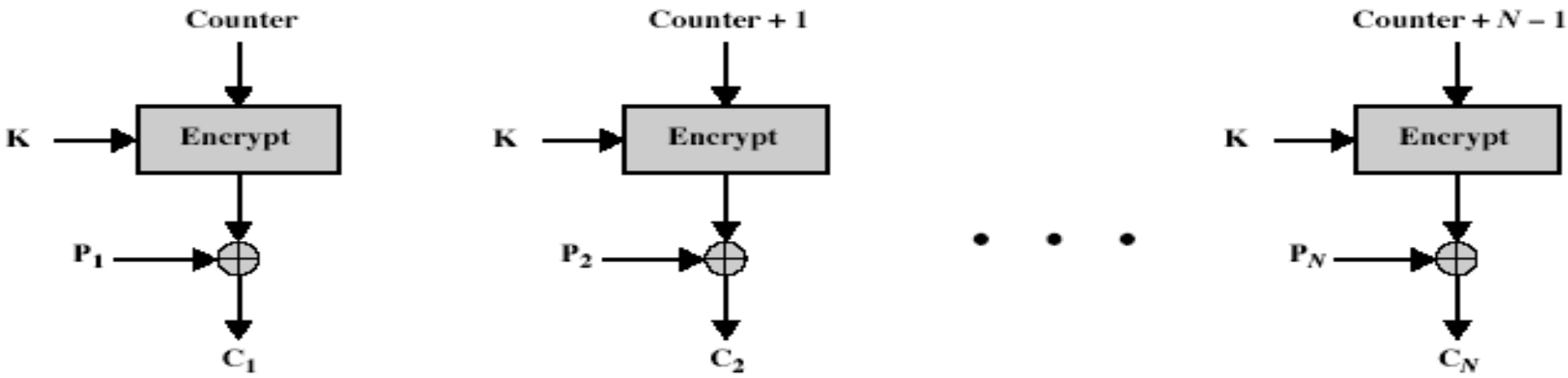
Advantages and Limitations of OFB

- used when error feedback a problem or where need to encryptions before message is available
- superficially similar to CFB
- but feedback is from the output of cipher and is independent of message
- sender and receiver must remain in sync, and some recovery method is needed to ensure this occurs

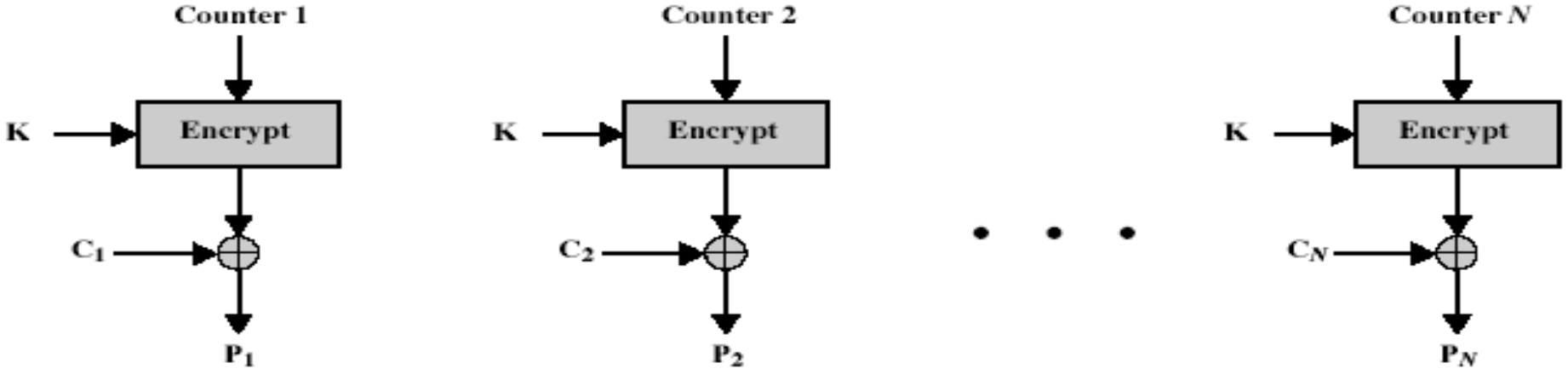
Counter Mode (CTR) 1/2

- 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)
- uses: high-speed network encryptions

Counter Mode (CTR) 2/2



(a) Encryption



(b) Decryption

Advantages and Limitations of CTR

- efficiency
  - can do parallel encryptions
  - in advance of need
  - good for bursty high speed links
- random access to encrypted data blocks
- provable security (good as other modes)
- but must ensure never reuse key/counter values, otherwise could break (cf OFB)

Triple DES

- clear a replacement for DES was needed
  - 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

Why Triple-DES?

- why not Double-DES?
  - $C = E_{K2}[E_{K1}[P]]$
  - NOT same as some other single-DES use, but have (can find $K3$ such that $C = E_{K3}[P]$)
- 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 to break

Triple-DES with Two-Keys

- hence must use 3 encryptions
  - would seem to need 3 distinct keys
- but can use 2 keys with E-D-E sequence
  - $C = E_{K1}[D_{K2}[E_{K1}[P]]]$
  - nb encrypt & decrypt equivalent in security
  - if K1=K2 then can work with single DES
- standardized in ANSI X9.17 & ISO8732
- no current known practical attacks

Triple-DES with Three-Keys

- although are no practical attacks on two-key Triple-DES have some indications
- can use Triple-DES with Three-Keys to avoid even these
  - $C = E_{K3}[D_{K2}[E_{K1}[P]]]$
- has been adopted by some Internet applications, e.g. PGP, S/MIME