Design of
Modern Block Ciphers

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

Teacher: Dipanwita Roy Chowdhury

Shannon's Main Contributions

- Notions of theoretical security and practical security
- Observation that the secret is all in the key, not in the algorithm
- **Product ciphers** and **mixing transformations** – inspiration for **DES, AES and …………**
- Proof that Vernam's cipher (one-time pad) was theoretically secure

Product Cryptosystems

- To use two or more cryptosystems is to encrypt and decrypt messages consecutively. We call this a **product cipher**.

- He believes that a combination of an initial transposition (Permutation) with alternating substitutions and linear operations may do the trick.

- Both DES and AES use Shannon's ideas of Product System and of type Substitution Permutation Network (SPN).

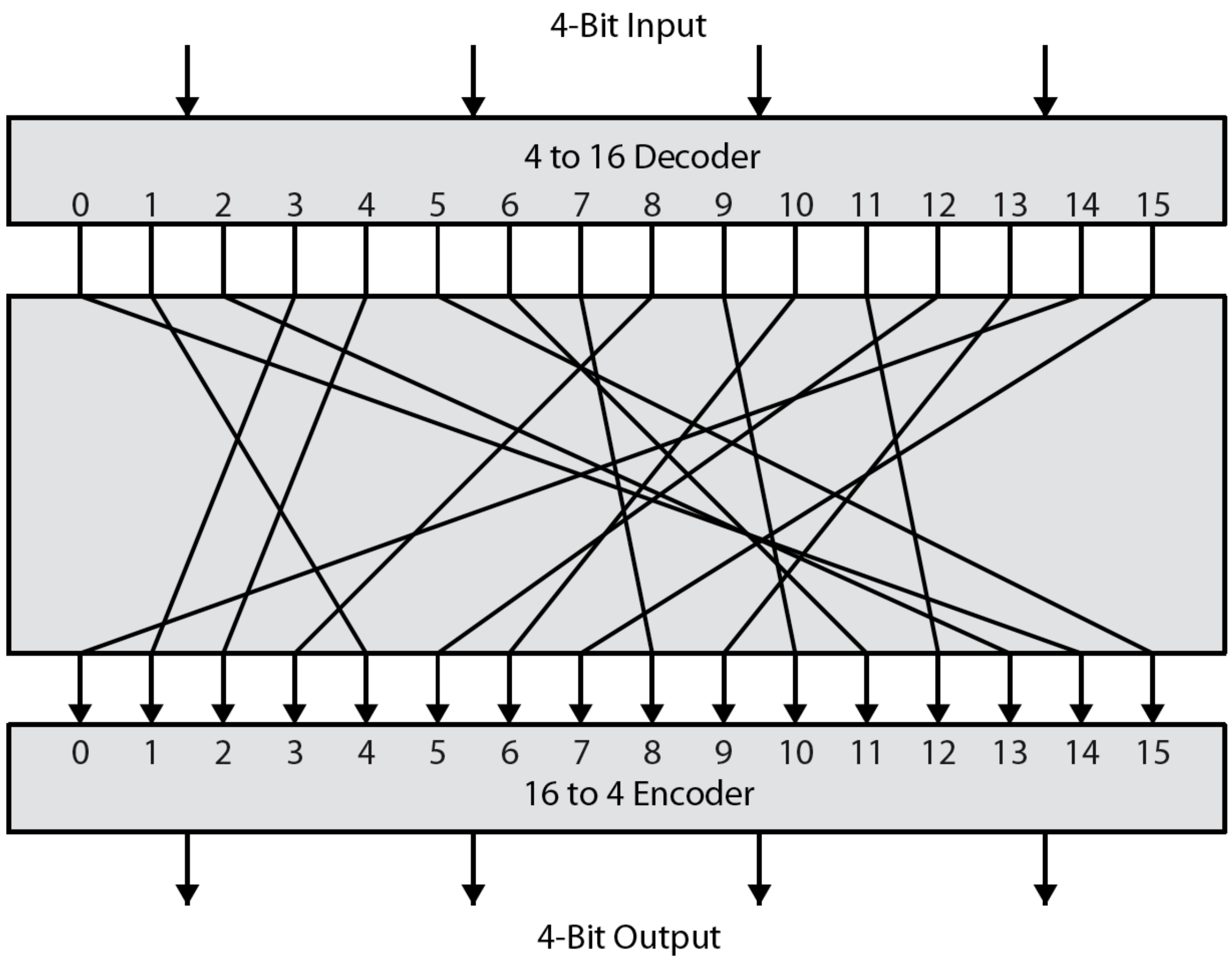
Block vs Stream Ciphers

- block ciphers process messages in blocks, each of which is then en/decrypted
- like a substitution on very big characters
  - 64-bits or more
- stream ciphers process messages a bit or byte at a time when en/decrypting
- many current ciphers are block ciphers
- broader range of applications

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
- instead create from smaller building blocks
- using idea of a product cipher

Ideal Block Cipher



Claude Shannon and Substitution-Permutation Ciphers

- Claude Shannon introduced idea of substitution-permutation (S-P) networks in 1949 paper
- form basis of modern block ciphers
- S-P nets are based on the two primitive cryptographic operations seen before:
  - *substitution* (S-box)
  - *permutation* (P-box)
- provide *confusion* & *diffusion* of message & key
  - diffusion – dissipates statistical structure of plaintext over bulk of ciphertext
  - confusion – makes relationship between ciphertext and key as complex as possible

Diffusion and Confusion

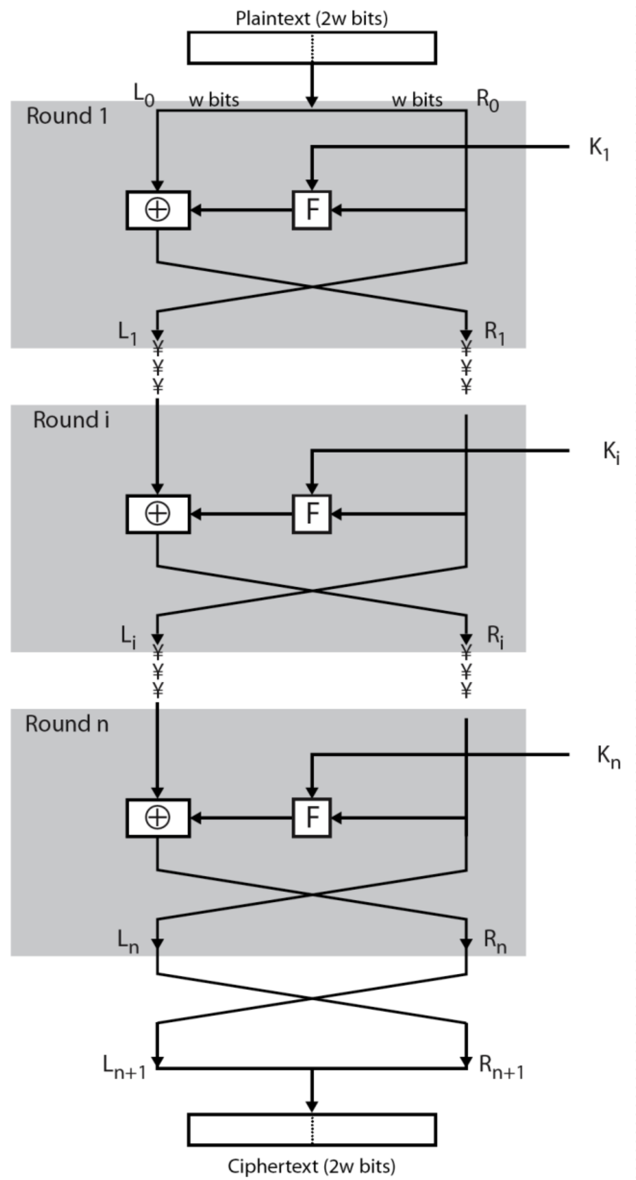
- Diffusion – dissipates statistical structure of plaintext over bulk of ciphertext
- confusion – makes relationship between ciphertext and key as complex as possible
- Encryption function E, C = E (K,P)

Relation of (1) C, P   (ii) C, K

Feistel Cipher Structure

- 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 S-P net concept

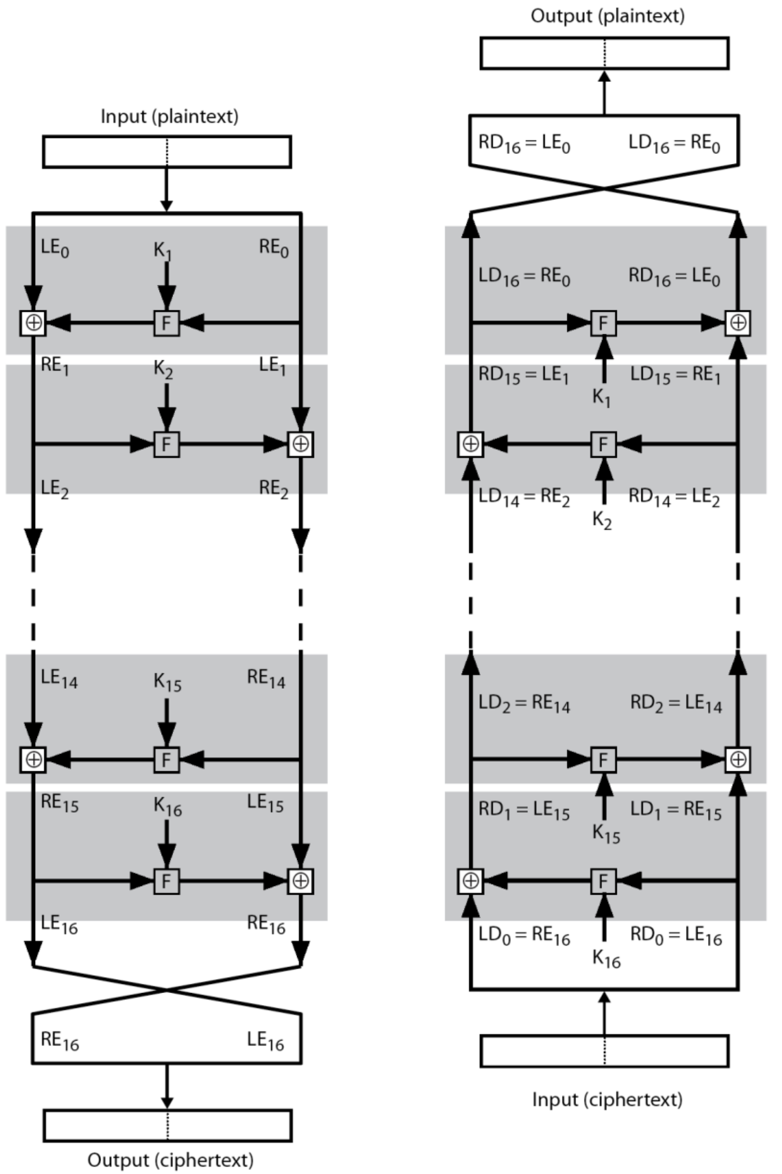
Feistel Cipher Structure



Feistel Cipher Design Elements

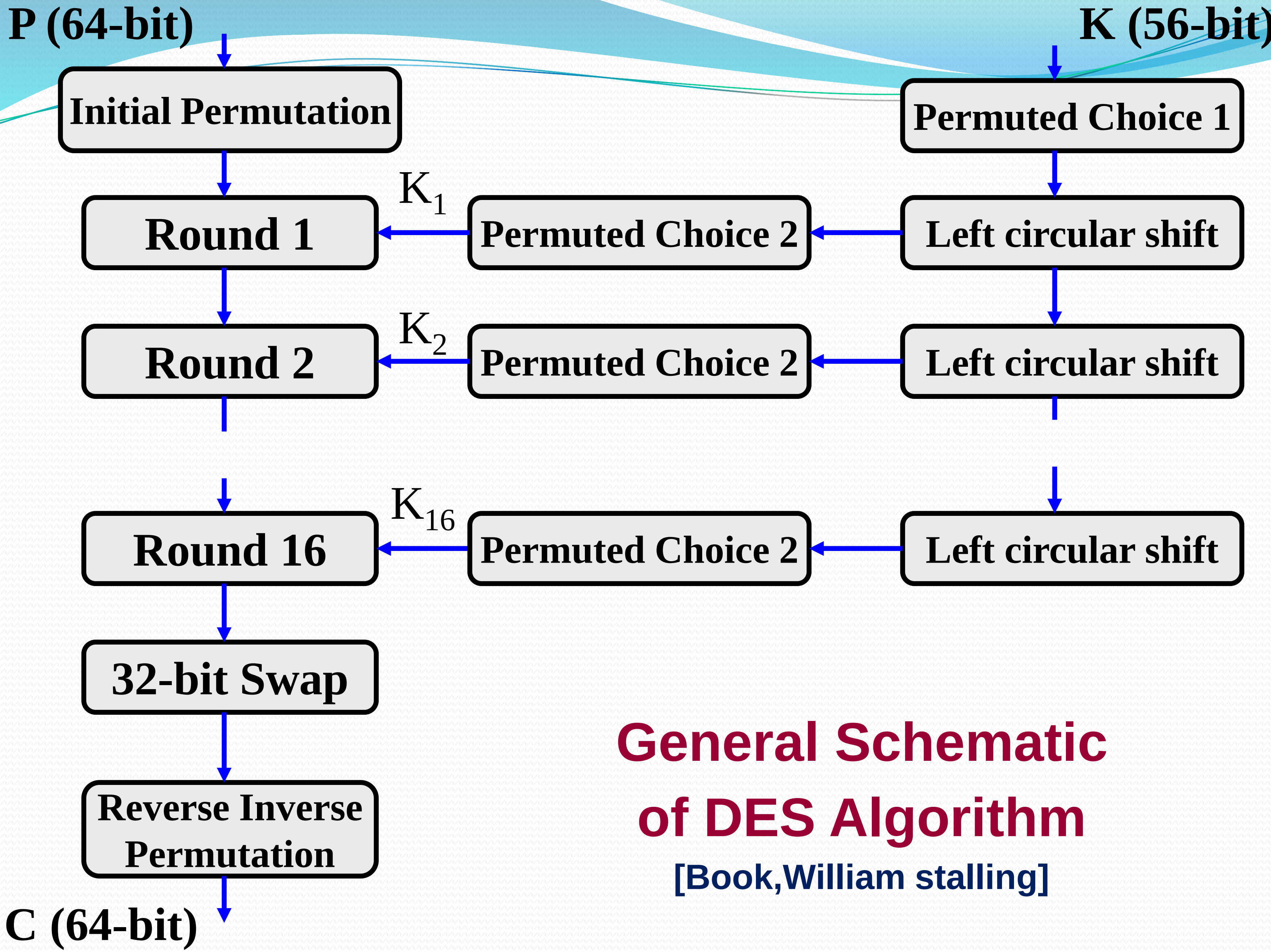
- block size
- key size
- number of rounds
- subkey generation algorithm
- round function
- fast software en/decryption
- ease of analysis

Feistel Cipher Decryption



Data Encryption Standard (DES)

- most widely used block cipher in world
- adopted in 1977 by NBS (now NIST)
  - as FIPS PUB 46
- encrypts 64-bit data using 56-bit key
- has widespread use
- has been considerable controversy over its security



General Schematic of DES Algorithm

[Book,William stalling]

Initial Permutation IP

- first step of the data computation
- IP reorders the input data bits
- even bits to LH half, odd bits to RH half
- quite regular in structure (easy in h/w)
- example:

```
IP(675a6967 5e5a6b5a) = (ffb2194d
004df6fb)
```

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

- the 1st bit of the output is taken from the 58th bit of the input; the 2nd bit from the 50th bit, and so on, with the last bit of the output taken from the 7th bit of the input.

Inverse IP

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

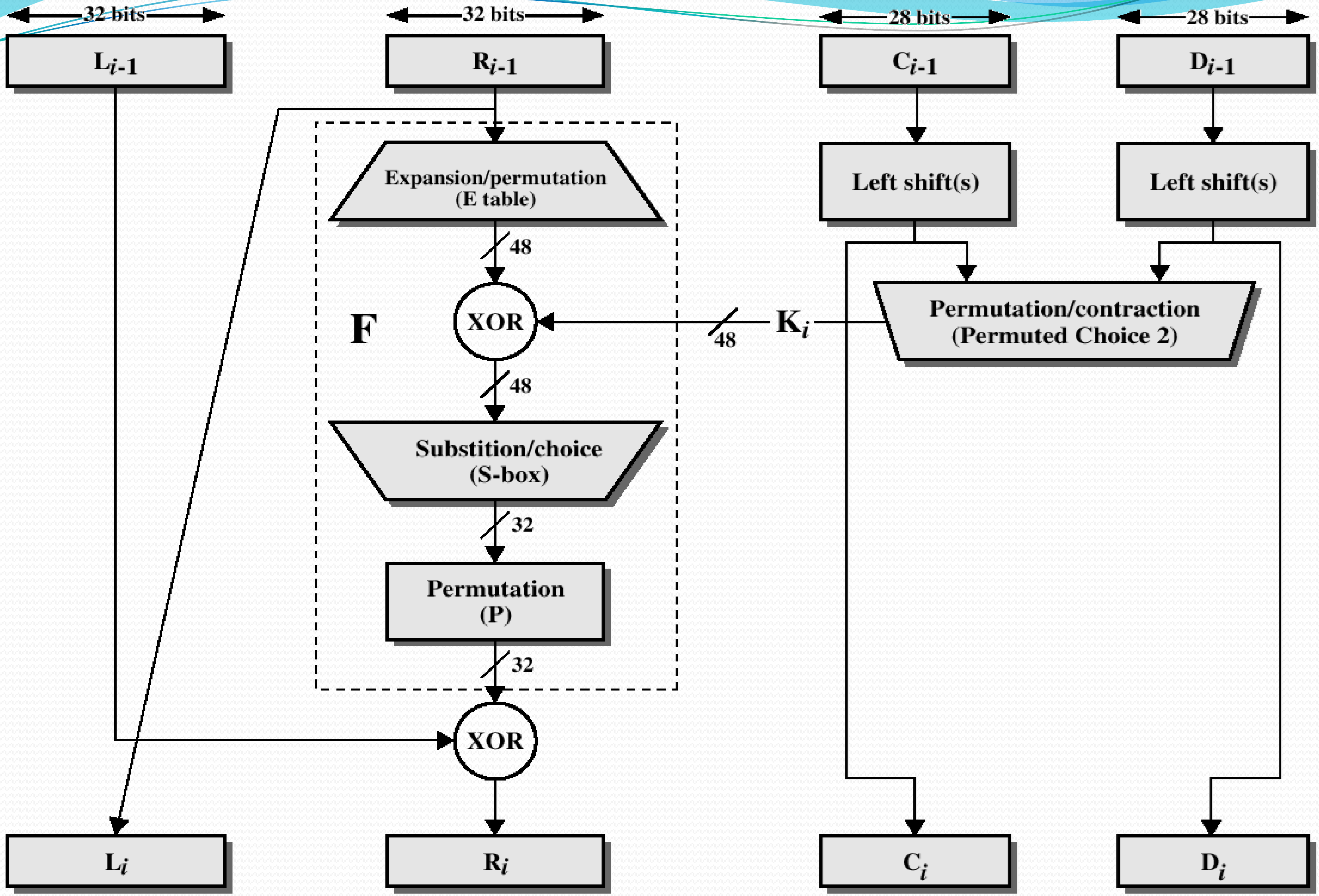
33 1 41 9 49 17 57 25

DES Round Structure

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

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

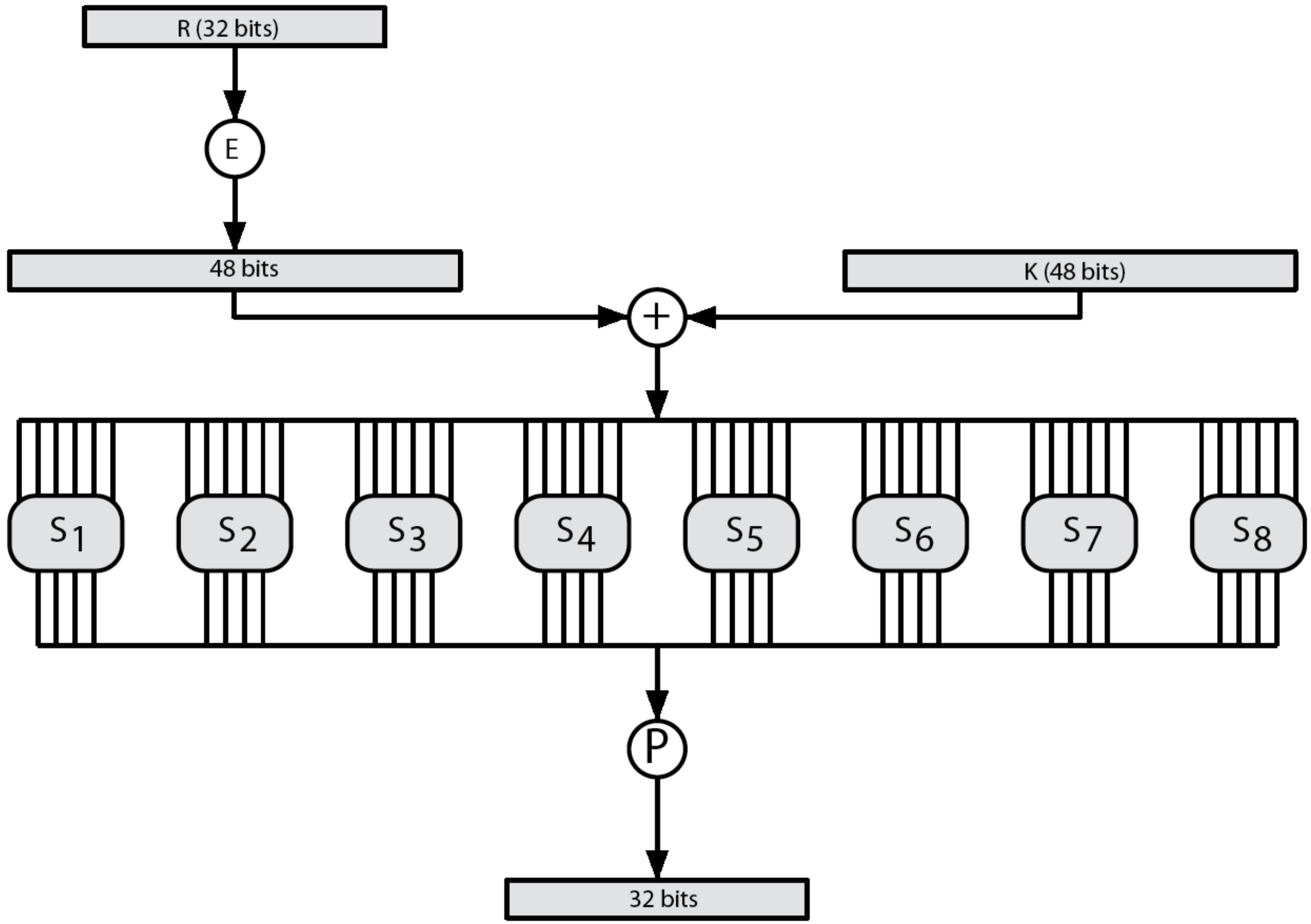
- F takes 32-bit R half and 48-bit subkey:
  - expands R to 48-bits using perm E
  - adds to subkey using XOR
  - passes through 8 S-boxes to get 32-bit result
  - finally permutes using 32-bit perm P



Single Iteration of DES Algorithm

[Book,William stalling]

DES Round Structure



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

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 of 4
  - inner bits 2-5 (**col** bits) are substituted
  - result is 8 lots of 4 bits, or 32 bits
- row selection depends on both data & key
  - feature known as autoclaving (autokeying)
- example:
  - `S(18 09 12 3d 11 17 38 39) = 5fd25e03`

DES S-box

- 8 “substitution boxes” or S-boxes, S1, S2.... S_8
- For each S-box, input: 6 bits, output: 4 bits

Example:

S-box: S matrix 4 x 16, values 0 to 15

Input: 6 bits b1 b2 b3 b4 b5 b6

Row address r: 2 bits(b1 b6),

Column address c: 4 bits(b2 b3 b4 b5)

Output: binary representation of S(r,c)

DES S-box

- Example S-Box S1

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
6	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	9
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	15

- Input: **111001**, Output: **1010** [row 3 (11), column 12 (1100)]

```
   | 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111
-----------------------------------------------------------------------------------
00 | 1110 0100 1101 0001 0010 1111 1011 1000 0011 1010 0110 1100 0101 1001 0000 0111
01 | 0000 1111 0111 0100 1110 0010 1101 0001 1010 0110 1100 1011 1001 0101 0011 1000
10 | 0100 0001 1110 1000 1101 0110 0010 1011 1111 1100 1001 0111 0011 1010 0101 0000
11 | 1111 1100 1000 0010 0100 1001 0001 0111 0101 1011 0011 1110 1010 0000 0110 1101
```

DES Key Schedule

- forms subkeys used in each round
  - initial permutation of the key (PC1) which selects 56-bits in two 28-bit halves
  - 16 stages consisting of:
    - rotating **each half** separately either 1 or 2 places depending on the **key rotation schedule** K
    - selecting 24-bits from each half & permuting them by PC2 for use in round function F
- note practical use issues in h/w vs s/w

DES Decryption

- decrypt must unwind steps of data computation
- with Feistel design, do encryption steps again using subkeys in reverse order (K16 … K1)
  - IP undoes final FP step of encryption
  - 1st round with K16 performs 16th encrypt round
  - ....
  - 16th round with K1 performs 1st encrypt round
  - then final FP undoes initial encryption IP
  - thus recovering original data value

DES Avalanche Effect

- key desirable property of encryption
- where a change of **one** input or key bit results in changing approx **half** output bits
- making attempts to "home-in" by guessing keys impossible
- DES exhibits strong avalanche

DES Avalanche Effect

Change in Plaintext **Change in Key**

Round	No. of bits that differ		Round	No. of bits that differ
0	1		0	0
1	6		1	2
2	21		2	14
3	35		3	28
4	39		4	32
5	34		5	30
6	32		6	32
7	31		7	35
8	29		8	34
.			.	
.			.	
16	34		16	35

Cryptanalysis of DES

DES Weak Keys

- DES uses 16 48-bits keys generated from a master 56-bit key (64 bits if we consider also parity bits, every 8th bit is parity bit)
- • Weak keys: keys make the same sub-key to be generated in more than one round.
- • Result: reduce security
- • DES has 4 weak keys (64 bits)

  (i) 01010101 01010101

  (ii) FEFEFEFE .... FEFEFEFE

  (iii) E0E0E0E0 .... E0E0E0E0

  (iv) 1F1F1F1F 1F1F1F1F

- If all the sub-keys are identical then encryption function becomes self inverting and using two encryption original plaintext can be found.

$E_K(E_K(x))=x$, since encryption and decryption are same.

Semi Weak Keys DES

- DES has also semi-weak keys, which only produce two different subkeys, each used eight times in the algorithm
- If K1 and K 2 are two such sub-keys, then

  they have the property that $E_{K1}(E_{K2}(x))=x$
- There are six pairs of DES semi-weak keys •
- Weak and semi-weak keys are not considered "flaws" of DES. There are 256 (7.21×10^{16}) possible keys for DES, of which only four are weak and twelve are semi-weak ...

Brute Force Attacks on DES

- Known-Plaintext Attack (several pairs of plaintext-ciphertext are known)
- Try all 2^{56} (= 7.2×10^{16}) possible keys
- brute force search looks hard
- recent advances have shown is possible
  - in 1997 on Internet in 3 months
  - in 1998 on dedicated h/w (EFF) in 3 days
  - in 1999 above combined in 22hrs!

Strength of DES: Analytic Attacks

- now have several analytic attacks on DES
- these utilize some deep structure of the cipher
  - by gathering information about encryptions
  - can eventually recover some/all of the sub-key bits
  - if necessary then exhaustively search for the rest
- generally these are statistical attacks
- include
  - differential cryptanalysis
  - linear cryptanalysis
  - related key attacks

DES Design Criteria

- as reported by Coppersmith in [COPP94]
- 7 criteria for S-boxes provide for
  - non-linearity
  - resistance to differential cryptanalysis
  - good confusion
- 3 criteria for permutation P provide for
  - increased diffusion

Block Cipher Design

- basic principles still like Feistel's in 1970's
- number of rounds
  - more is better, exhaustive search best attack
  - But should be cost-efficient
- function f:
  - provides "confusion", is nonlinear, avalanche
  - have issues of how S-boxes are selected
- key schedule
  - complex subkey creation, key avalanche
  - Key should be random