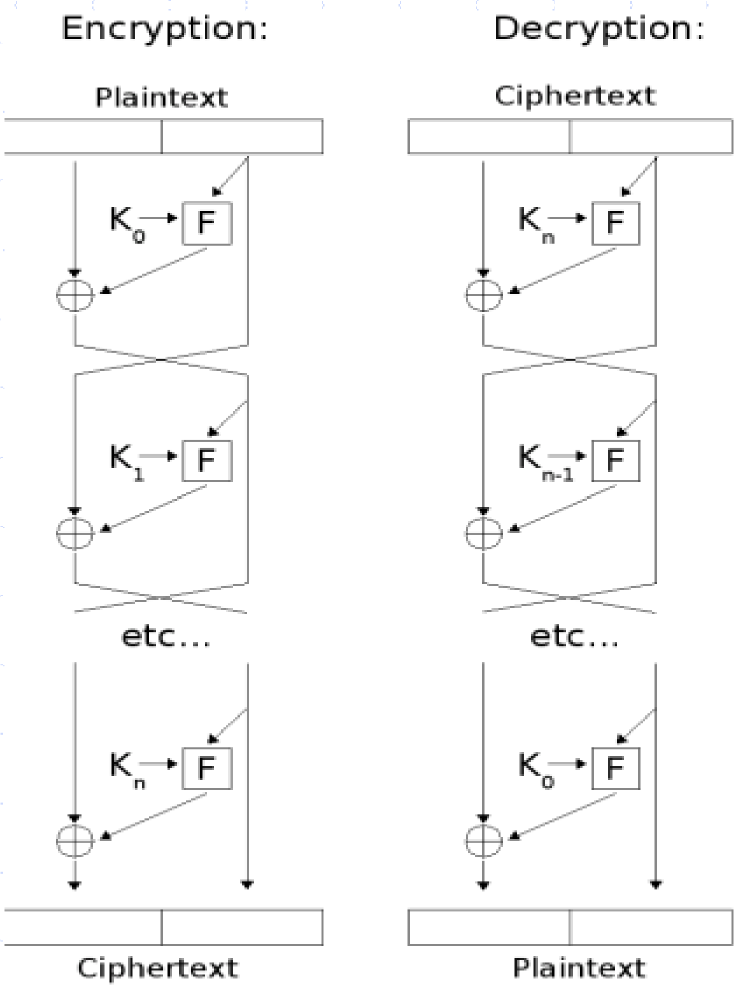
DES

Overview

- DES was first published on March 17, 1975
  - as a result of solicitation of cryptosystems by NIST on May 15, 1973.
- Adopted as a standard for “unclassified” applications on Jan 15, 1977.
  - was initially expected to be used for 10-15 years as a standard, but
    - lasted for more
  - was reviewed every five years after its adoption
  - eventually, was finally reviewed in 1999, after which AES replaced it.

Overview (contd)

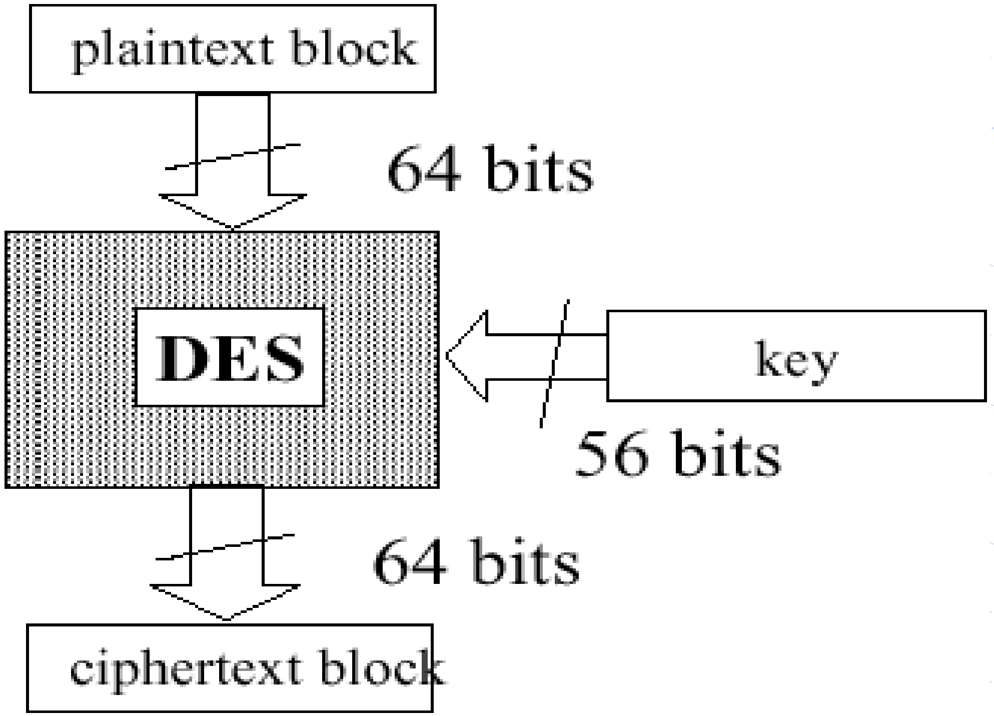
- DES
  - is a special type of iterated cipher….following Feistel design
- It is
  - a 16-round Feistel cipher with block length of 64
  - but uses a 56-bit key.
- There was controversy over its design
  - in choice of 56-bit key (vs Lucifer 128-bit, on which it was based upon)
  - design criteria (of the S-boxes) were classified



Feistel Cipher

DES Overview

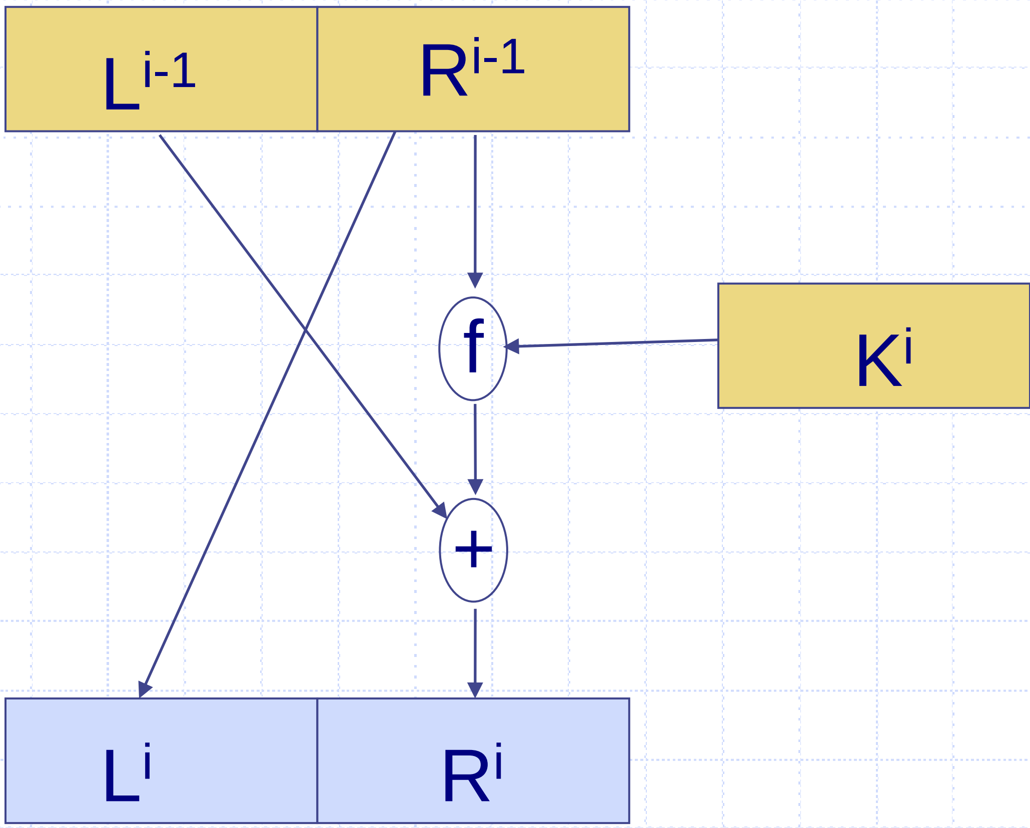
- encrypts 64-bit data using a 56-bit key
- keys
  - 64 but actually 56 bits
  - every 8^{th} bit is parity and is ignored
  - are LSBs of the key bytes
  - some keys are weak keys
    - 64 out of 72,057,594,037,927,927,936
- blocks
  - 64 bit blocks in/out
  - composed of bits numbered from left to right, i.e., the left most bit of a block is bit one.



DES Algorithm

- Algorithm Description
  - Start with 64 bits of Plaintext
  - Initial Permutation (IP)
    - permute 64-bit block according to IP i.e. $IP(x) = L^0R^0$
  - Block bisected into two 32 bits "right" and "left" blocks and Feistel network function applied
    - each
      - L^i and R^i – 32 bits in length
      - K^i - 48-bits in length
    - the Fesitel function f, therefore appears as

$$f: \{0,1\}^{32} \text{ X } \{0,1\}^{48} \rightarrow \{0,1\}^{32}$$

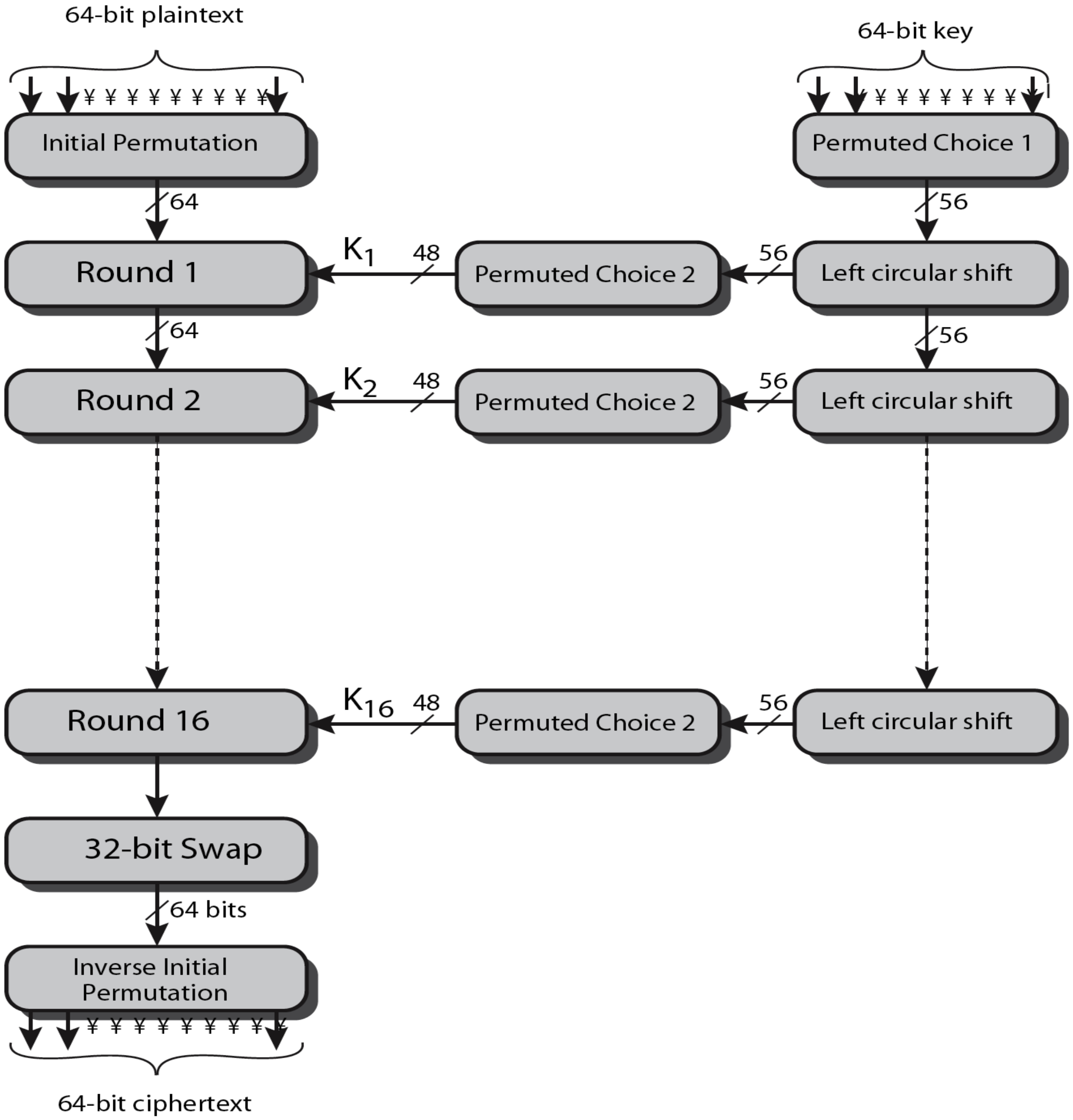


DES Algorithm

- Algorithm Description
  - At the end of 16th round L and R are reversed
    - Permutation using IP^{-1} which is the inverse of IP is done
  - End with 64 bits of Ciphertext

    i.e. $y = IP^{-1}(R^{16}L^{16})$
    - note that R and L sub-halves are interchanged here

DES Overview

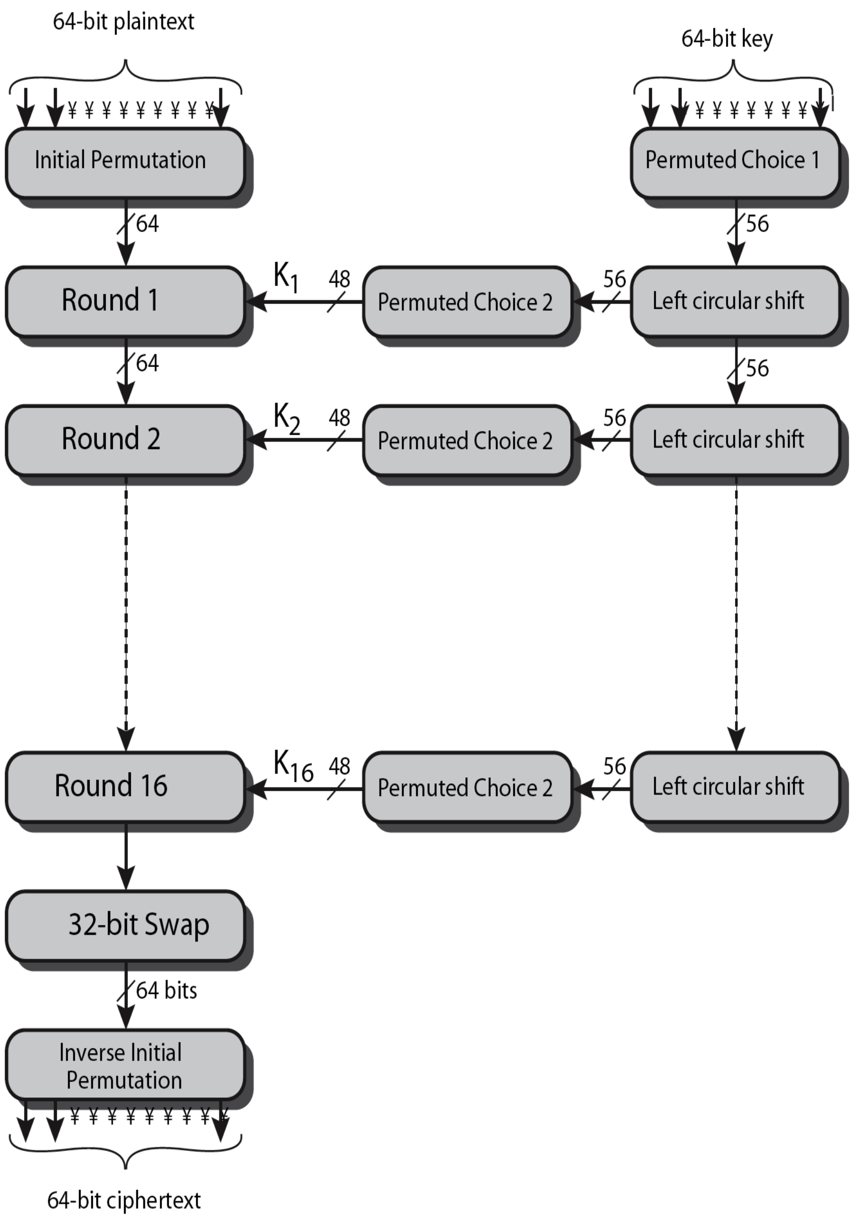


To Investigate

- Two aspects to be investigated further, with reference to previous figure
  - How the 56 bit keys are converted to 48 bits ?
  - How the Feistel function f works ?

DES Key schedule overview

- key is represented by 64-bits, however only 56 are used.
- key schedule overview
  - permute the key according to PC-1 (result is 56 bits)
  - split the key into two halves L and R
  - loop (16 times)
    - shift L and R left 1 OR 2 bits based on round number
    - now use L and R to create sub-key by permuting according to PC-2 (which only uses 48 of the 56 bits)
- each sub-key is 48-bits long



DES Key schedule overview

- Let the I/P 64-bit keys be as follows:

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16

17 ………………………………… 24

25………………………………… 32

33…………………………………40

41…………………………………48

49…………………………………56

57…………………………………64

- i.e. bits 8,16,24,……are dropped to reduce 64-bits to 56-bits

DES Key Schedule – PC1

- forms subkeys used in each round
  - initially, 64-bit DES key is reduced to a 56-bit key
    - ignoring every eight bit
    - used only for parity check to ensure key is error-free
    - this occurs as per the partial table (PC-1)below

(b) Permuted Choice One (PC-1)

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

    - 14 columns, four rows i.e. 56-bit key from 64-bit key

DES Key Schedule (contd)

- next, the 56-bit key is divided into two 28-bit halves
- each bit is shifted left by
  - either one or two bits depending on the round number
  - i.e.

round no	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
one/two bit	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

DES Key Schedule – PC2(contd)

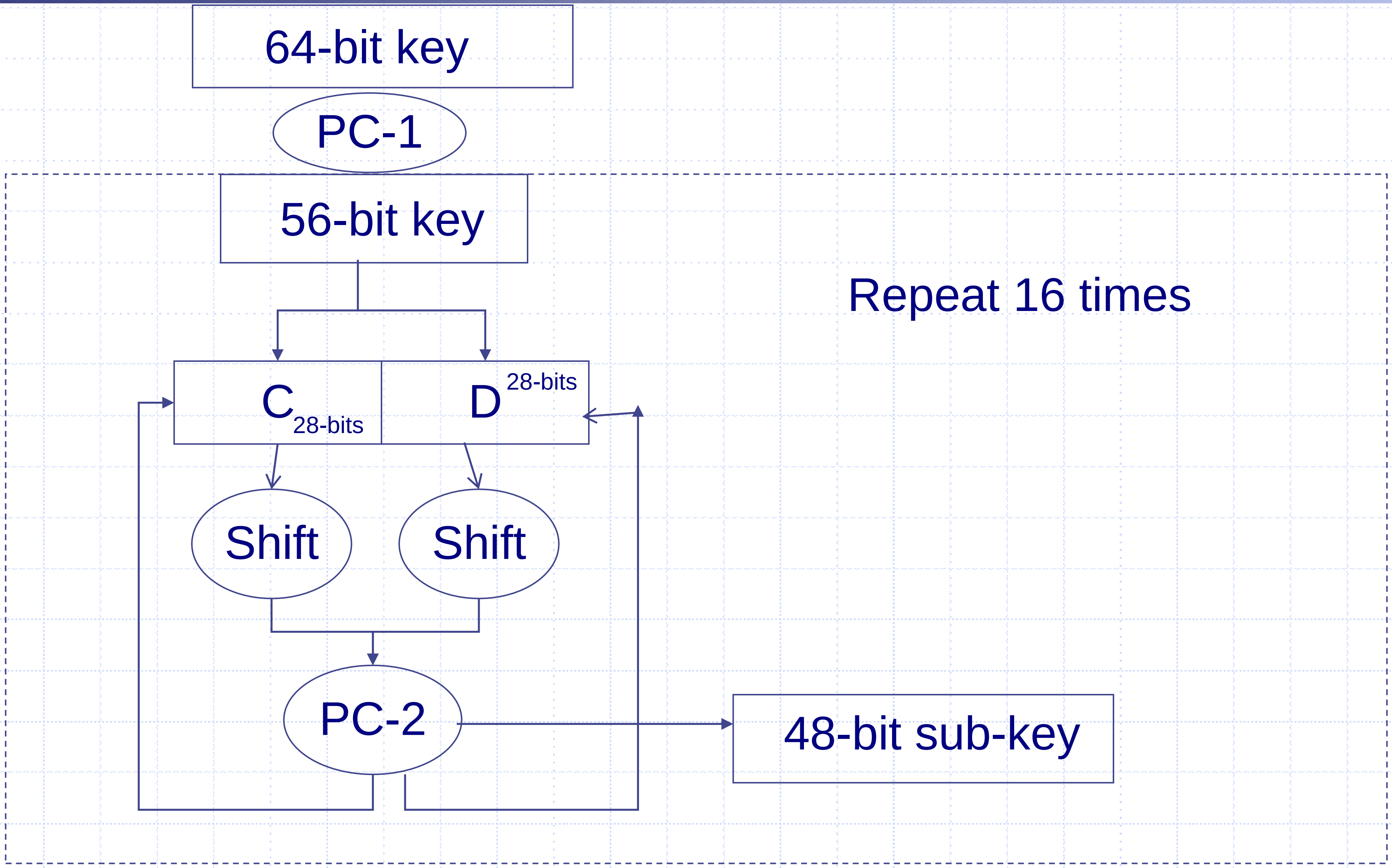
- Subkeys generation
  - next, 48-bit subkey is generated for each of the 16 rounds
    - 48 out of 56 bits are selected based on PC-2 shown below
    - also called compression permutation

(c) Permuted Choice Two (PC-2)

14	17	11	24	1	5	3	28
15	6	21	10	23	19	12	4
26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

- note practical use issues in h/w vs s/w

DES Key schedule - summary



DES/TDEA Initial Permutation

- Initial Permutation (IP) – *permutation expansion*
  - denoted as `IP(x) = L`^{0}`R`^{0}
  - reorders the input data bits
  - even bits to LH half, odd bits to RH half
  - quite regular in structure (easy in h/w)

- Transposes the input block using table

  i.e. Bit 58 goes to position 1, 50 to 2, 42 to 3, …

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

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

DES/TDEA Initial Permutation

- the DES initial and the final permutation
  - do not affect DES's security
  - why are these present in DES standard ?
- are often eliminated in software implementations
  - result no longer adheres DES standard

DES Round Structure – function F

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

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

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

- F takes 32-bit R-half and 48-bit subkey
  - expands R-half to 48-bits using a defined permutation
  - provides a longer result that can be compressed subsequently
  - but, the main cryptographic purpose is
    - introduces avalanche effect - allows one bit to affect two substitutions thereby creating a rapidly increasing dependency of output to input bits
  - shown in the figures on next two slides

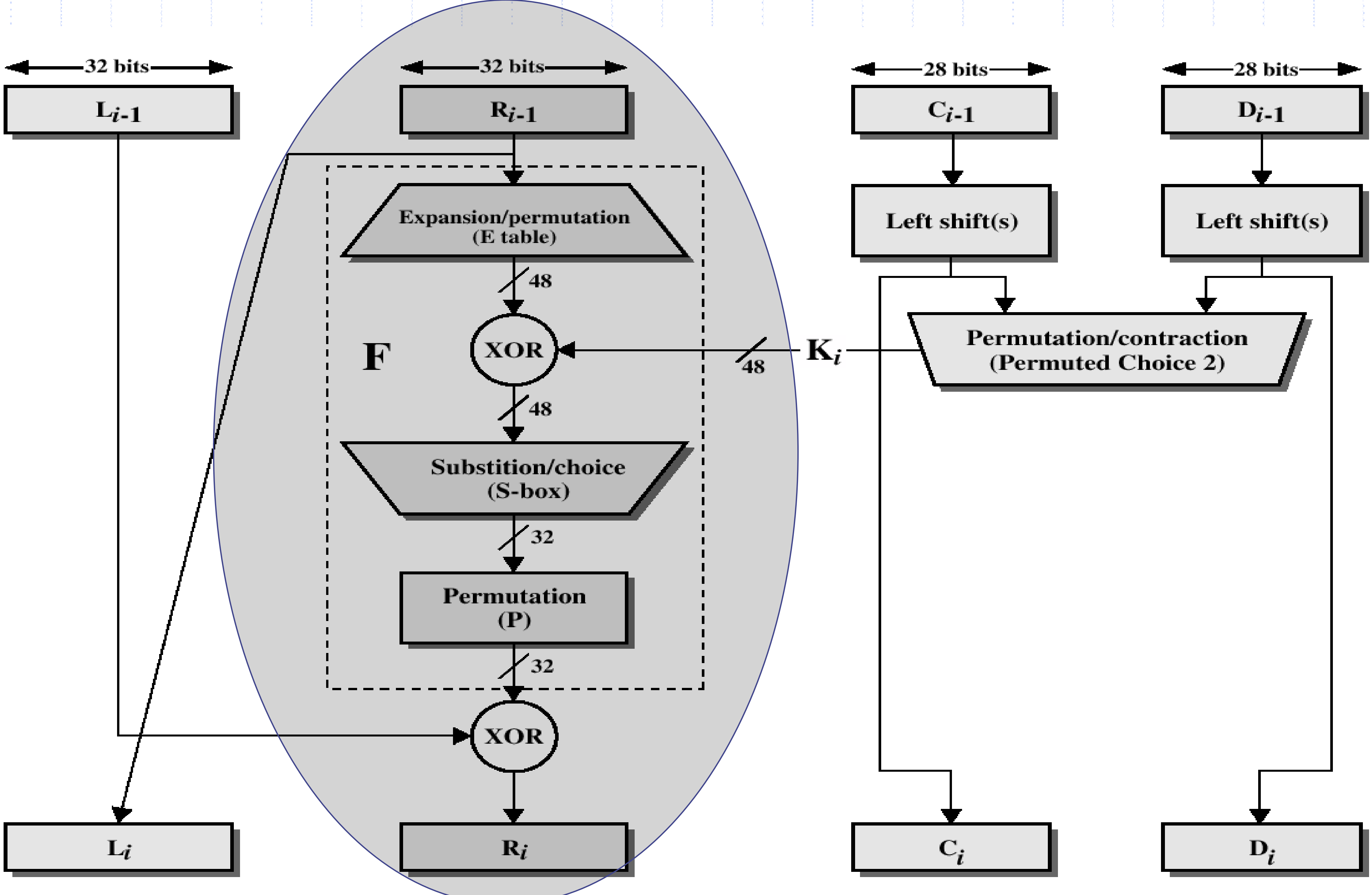
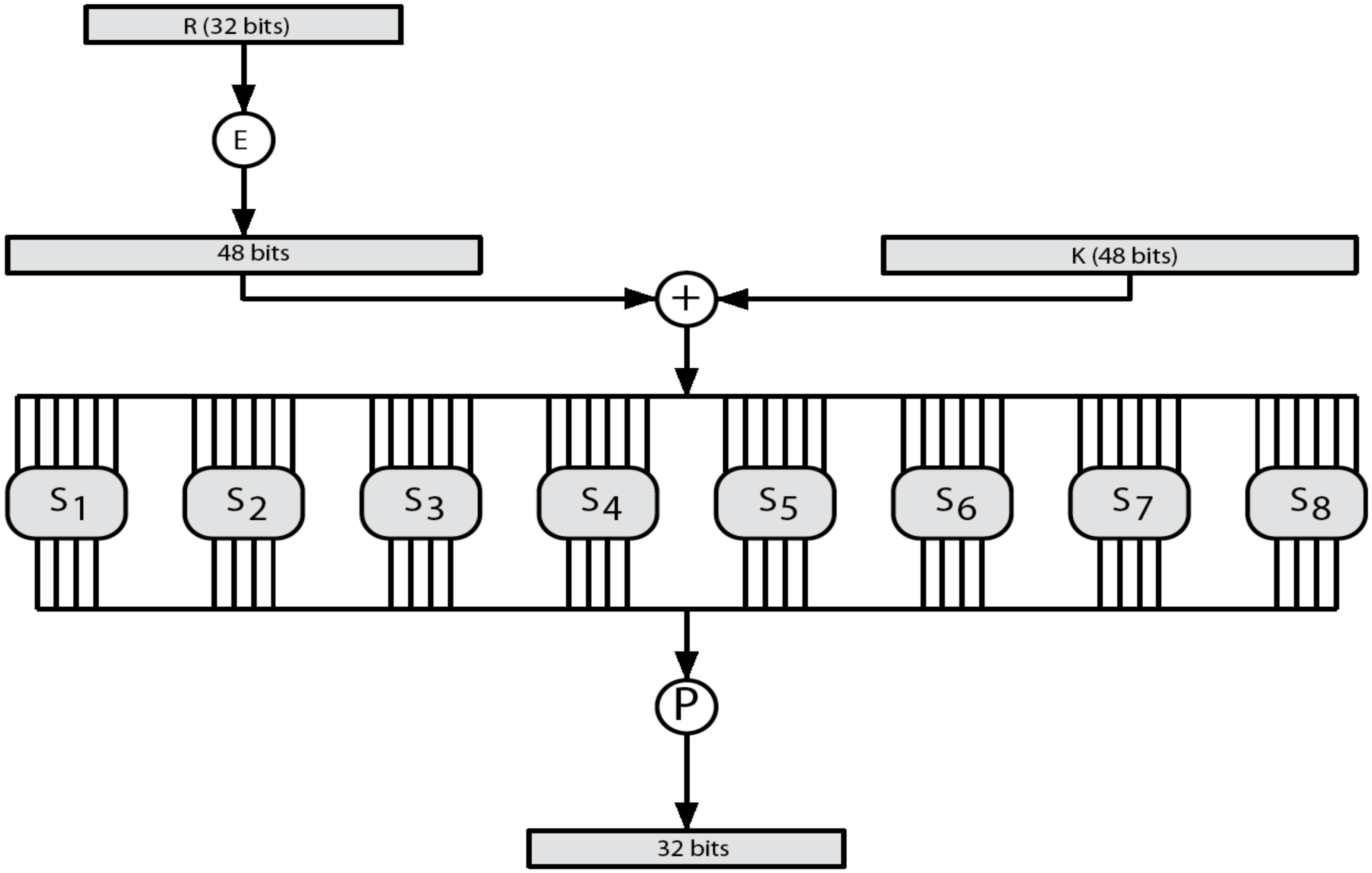


Figure 2.4 Single Round of DES Algorithm

DES Round Structure with S-box substitution



Expansion Permutation E(A) – the E-box

- Expansion function E(A) – A is 32-bits, output 48-bits

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

DES Round Structure

- Next,
  - adds the 48-bit output of E-box to the subkey of the stage using XOR
- Next,
  - passes through 8 S-boxes to get 32-bit result
  - Substitution Box used
    - 6 bit in → 4 bits out
    - each is a table of 4 rows and 16 columns
    - 8 different S-Boxes (memory requirement ??)
    - tables are used in parallel
    - 48 bits in 6 bit groups go through 8 S-boxes giving 32 bits out

Sample S-boxes (Stallings)

S_1	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	0	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	0
	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S_2	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S_3	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S_4	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

Sample S-boxes (Stallings)

S_5	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S_6	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S_7	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S_8	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

DES Round Structure – S-boxes

- Input bits i.e. eight 6-bit subblocks
  - used to index into S-box
- from the input bits b1-b6 say, indexing occurs as
  - each b1 and b6 are used to index one out of four rows
  - each of b2-b5 are used to index one of sixteen columns
  - e.g. how would the bit sequence for the input to the sixth S-box viz. 110011 treated ? How is 011001 to S1 treated ?

DES Round Structure – S-boxes

- Input bits i.e. eight 6-bit subblocks
  - used to index into S-box
- from the input bits b1-b6 say, indexing occurs as
  - each b1 and b6 are used to index one out of four rows
  - each of b2-b5 are used to index one of sixteen columns
  - e.g. how would the bit sequence for the input to the sixth S-box viz. 110011 treated ? 14 = e_{16} How is 011001 to S1 treated ?

DES Round Structure – S-boxes

- Input bits i.e. eight 6-bit subblocks
  - used to index into S-box
- from the input bits b1-b6 say, indexing occurs as
  - each b1 and b6 are used to index one out of four rows
  - each of b2-b5 are used to index one of sixteen columns
  - e.g. how would the bit sequence for the input to the sixth S-box viz. 110011 treated ? 14 = e_{16} How is 011001 to S1 treated ? 9 = 9_{16}
- from eight S-boxes, thus the result is a 32 bit block
- this substitution is most critical
  - "*other operations are linear and easy to analyze, The S-boxes are nonlinear and, more than anything else, give DES its security*"

Find out S-box replacements….

- S1(011000) lookup row 00 col 1100 in S1 to get
  - 5
- S2(001001) lookup row 01 col 0100 in S2 to get
  - 15 = f in hex
- S3(010010) lookup row 00 col 1001 in S3 to get
  - 13 = d in hex
- S4(111101) lookup row 11 col 1110 in S4 to get
  - 2

DES Round Structure – P-box

- Next,
  - the eight 4-bit outputs are combined to get a single 32-bit output
- Next,
  - the P-box permutation is performed
  - the output of S-boxes is permuted using 32-bit permutation table 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
  - P maps each input bit to an output position
  - No bits used twice and no bits ignored (straight permutation)
- Next,
  - the result of P-box is XORed with left 32 bits of initial 64-bit block
  - L and R switched
- Go around again…
- Repeat 16 times

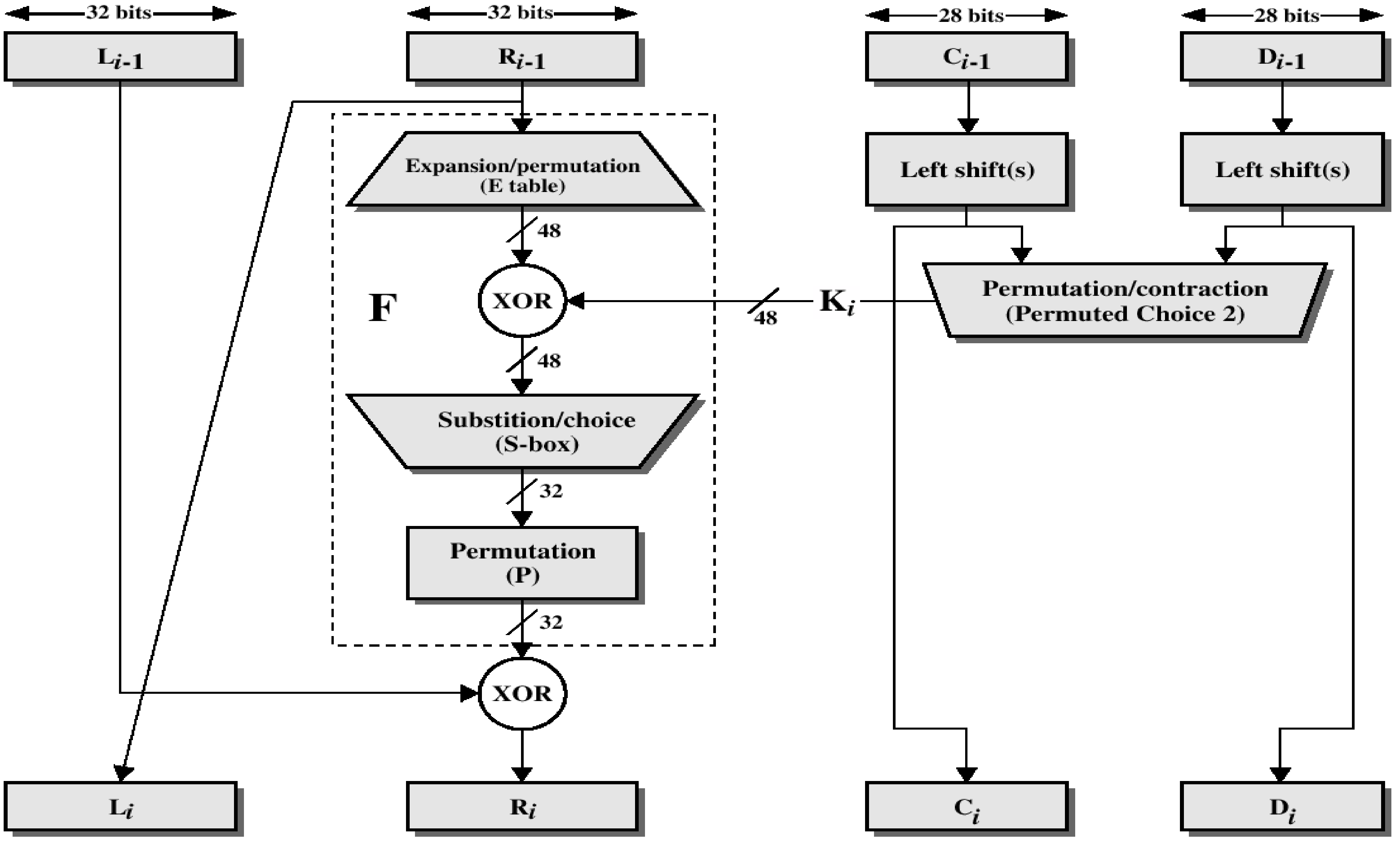


Figure 2.4 Single Round of DES Algorithm

DES Round Structure – Final Permutation

- Finally,
  - the final permutation is performed
  - inverse of the initial permutation

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

  - L and R not exchanged after last round
- Done

DEA/TDEA

DES Example from Stallings…

Table 3.5 DES Example

Round	K_i	L_i	R_i
IP		5a005a00	3cf03c0f
1	1e030f03080d2930	3cf03c0f	bad22845
2	0a31293432242318	bad22845	99e9b723
3	23072318201d0c1d	99e9b723	0bae3b9e
4	05261d3824311a20	0bae3b9e	42415649
5	3325340136002c25	42415649	18b3fa41
6	123a2d0d04262a1c	18b3fa41	9616fe23
7	021f120b1c130611	9616fe23	67117cf2
8	1c10372a2832002b	67117cf2	c11bfc09
9	04292a380c341f03	c11bfc09	887fbc6c
10	2703212607280403	887fbc6c	600f7e8b
11	2826390c31261504	600f7e8b	f596506e
12	12071c241a0a0f08	f596506e	738538b8
13	300935393c0d100b	738538b8	c6a62c4e
14	311e09231321182a	c6a62c4e	56b0bd75
15	283d3e0227072528	56b0bd75	75e8fd8f
16	2921080b13143025	75e8fd8f	25896490
IP^{-1}		da02ce3a	89ecac3b

DES Characteristics

- DES shows strong avalanche effect
  - one bit change input affects on average half of the output bits
  - to make attacks based on guessing difficult
- S-boxes are non-linear
  - provides confusion
    - i.e. makes relationship between ciphertext and key as complex as possible
  - What do we mean by non-linearity here ?
  - Why all other operations in DES are linear?

Security of DES – Key Size

- There have been other demonstrated breaks of the DES
  - 1993 key search chip design proposed by Weiner to crack in 1.5 days
  - 1997 on a large network of computers in a few months
  - 1998 on dedicated h/w (EFF) in a few days
  - 1999 above combined in 22hrs!
- All of the above are brute-force attack

Security of DES – Number of Rounds

- Why did DES use only 16 rounds and why not more or less ?
- DES round characteristics from published results
  - after every five rounds every ciphertext bit is a function of every plaintext bit
  - after every eight rounds, the ciphertext was a random function of every plaintext bit and every key bit

Security of DES – Number of Rounds

- Why did DES use only 16 rounds and why not more or less ?
- DES round characteristics from published results
  - after every five rounds every ciphertext bit is a function of every plaintext bit
  - after every eight rounds, the ciphertext was a random function of every plaintext bit and every key bit

called Avalanche effect