DATA ENCRYPTION STANDARD (DES)

Outline

- History
- Encryption
- Key Generation
- Decryption
- Strength of DES
- Ultimate

History

In 1971, IBM developed an algorithm, named LUCIFER which operates on a block of 64 bits, using a 128-bit key

Walter Tuchman, an IBM researcher, refined LUCIFER and reduced the key size to 56-bit, to fit on a chip.

History

In 1977, the results of Tuchman's project of IBM was adopted as the Data Encryption Standard by NSA (NIST).

A Simplified DES-Type Algorithm

- Suppose that a message has 12 bits and is written as L_0R_0 , where L_0 consists of the first 6 bits and R_0 consists of the last 6 bits.
- The key K has 9 bits. The *i*th round of the algorithm transforms an input $L_{i-1}R_{i-1}$ to the output L_iR_i using an 8-bit key K_i derived from K.
- The main part of the encryption process is a function $f(R_{i-1},K_i)$ that takes a 6-bit input

R_{i-1} and an 8-bit input K_i and produces a 6-bit output which will be described later.

The output of the *i*th round is defined as:

$$L_i = R_{i-1} \text{ and } R_i = L_{i-1} \text{ XOR } f(R_{i-1}, K_i)$$

The decryption is the reverse of encryption.

$[L_n]\ [R_n \text{ XOR } f(L_n, K_n)] = \ldots = [R_{n-1}]\ [L_{n-1}]$

The Operations of f Function

- $E(L_i)$=E(011001)=E(01010101) (Expander)
- S-boxes

S_1 101 010 001 110 011 100 111 000
001 100 110 010 000 111 101 011

S_2 100 000 110 101 111 001 011 010
101 011 000 111 110 010 001 100

The input for an S-box has 4 bits. The first bit specifies which row will be used: 0 for 1st

- The other 3 bits represent a binary number that specifies the column: 000 for the 1st column, 001 for the 2nd column, … 111 for the 7th column. For example, an input 1010 for S_1 box will yield the output 110.
- The key K consists of 9 bits. K_i is the key for the ith round starting with the ith bit of K. Let K=010011001, then K_4=01100101.

R_{i-1}=100110 and K_i=01100101

- $E(R_{i-1})$ XOR K_i =10101010 XOR 01100101

  = 11001111

S_1(1100)=000

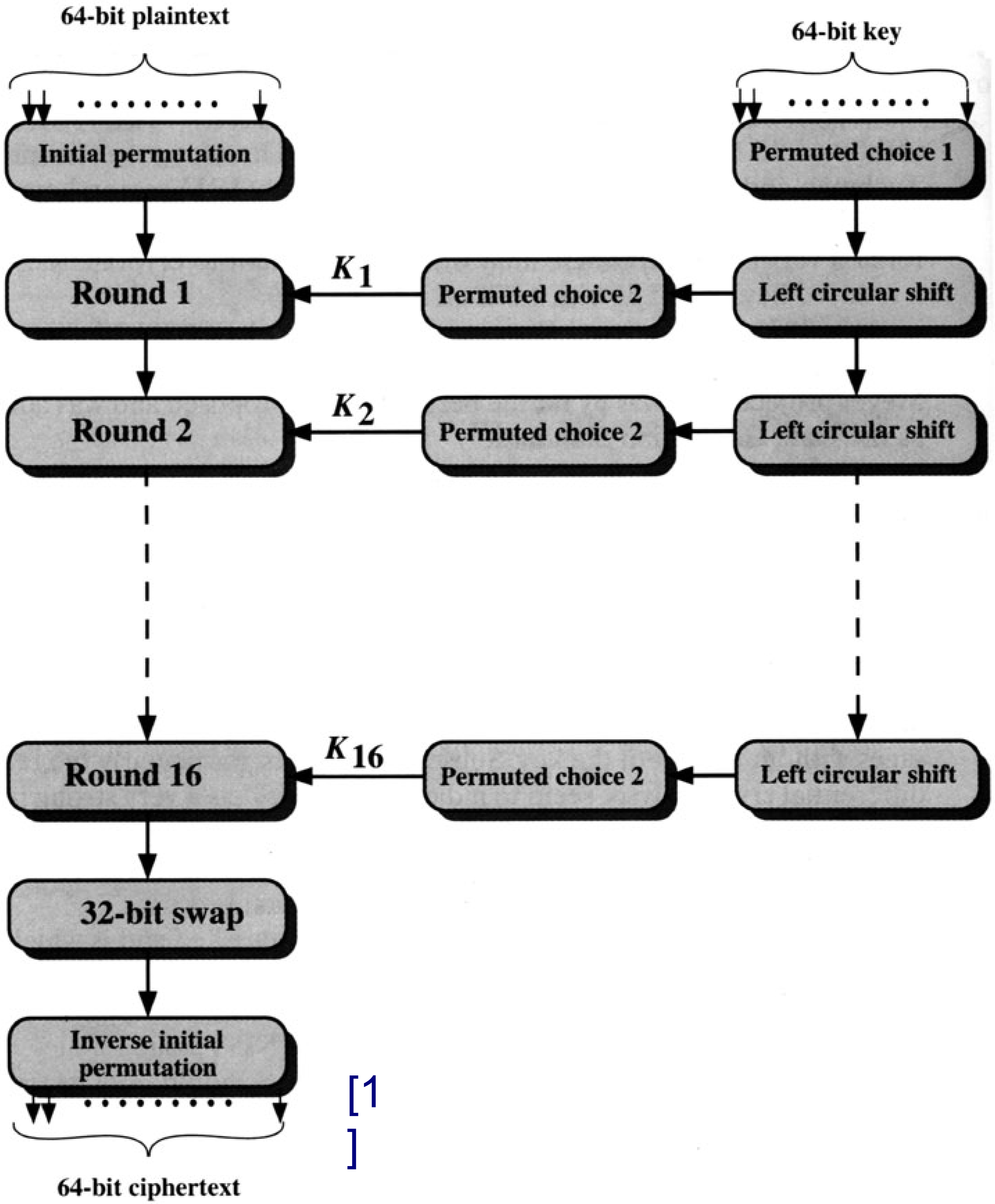
S_2(1111)=100

Thus, $R_i = f(R_{i-1},K_i)$=000100, $L_i = R_{i-1}$ =100110

$L_{i-1}R_{i-1}$ = 011100100110 → (?) L_iR_i

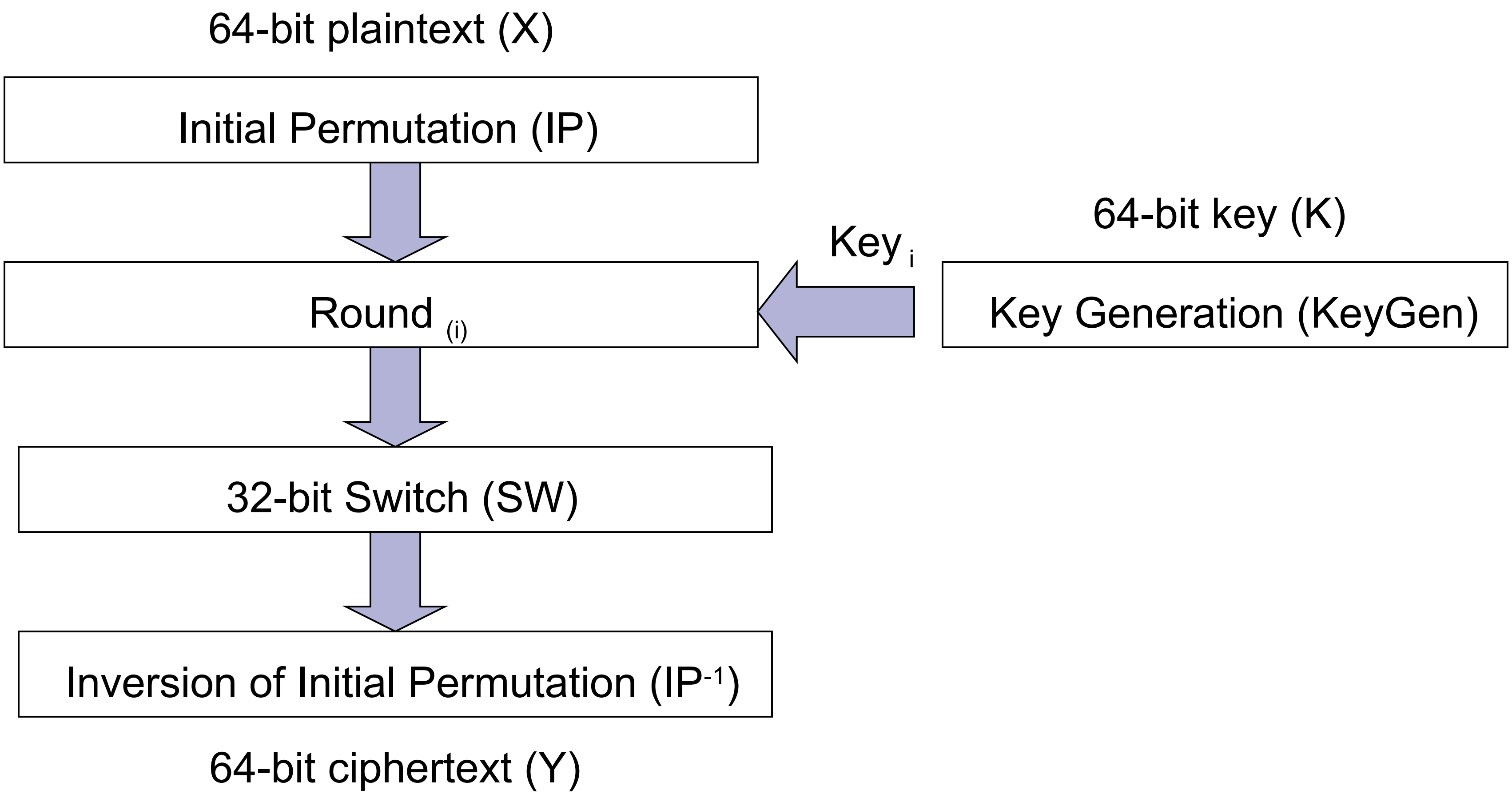
100110011000

Encryption



[1]

Encryption (cont.)



Encryption (cont.)

- Plaintext: X
- Initial Permutation: IP( )
- Round_i: $1 \le i \le 16$
- 32-bit switch: SW( )
- Inverse IP: IP^{-1}( )
- Ciphertext: Y
- $Y = IP^{-1}(SW(Round_i(IP(X), Key_i)))$

Encryption (IP, IP^{-1})

- IP

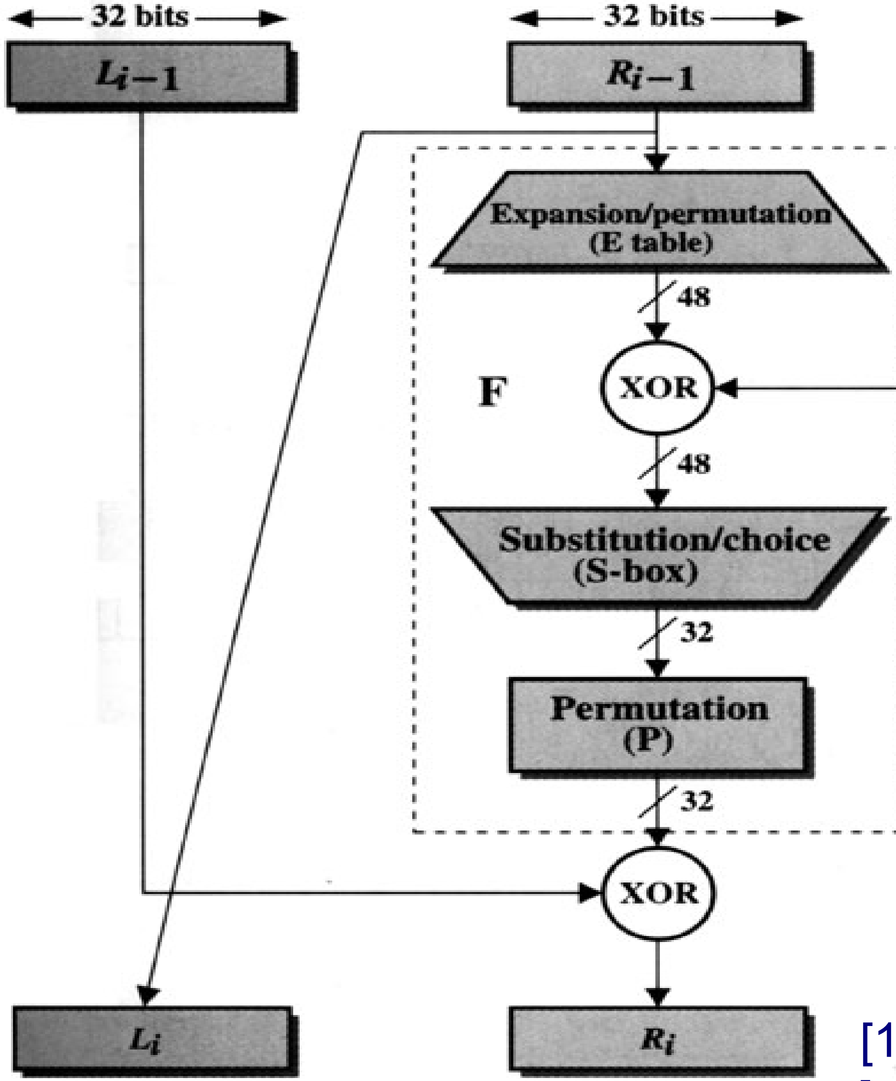
Bit	0	1	2	3	4	5	6	7
1	58	50	42	34	26	18	10	2
9	60	52	44	36	28	20	12	4
17	62	54	46	38	30	22	14	6
25	64	56	48	40	32	24	16	8
33	57	49	41	33	25	17	9	1
41	59	51	43	35	27	19	11	3
49	61	53	45	37	29	21	13	5
57	63	55	47	39	31	23	15	7

- IP^{-1}

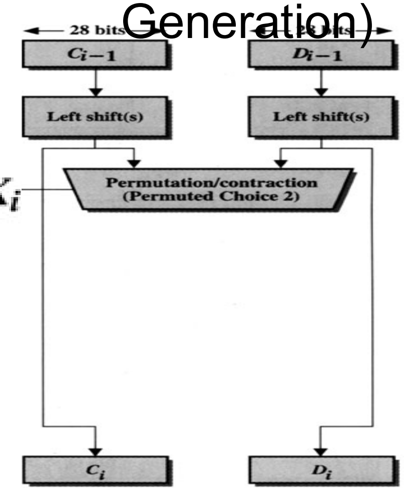
Bit	0	1	2	3	4	5	6	7
1	40	8	48	16	56	24	64	32
9	39	7	47	15	55	23	63	31
17	38	6	46	14	54	22	62	30
25	37	5	45	13	53	21	61	29
33	36	4	44	12	52	20	60	28
41	35	3	43	11	51	19	59	27
49	34	2	42	10	50	18	58	26
57	33	1	41	9	49	17	57	25

- Note: $IP(IP^{-1}) = IP^{-1}(IP) = I$

Encryption (Round)

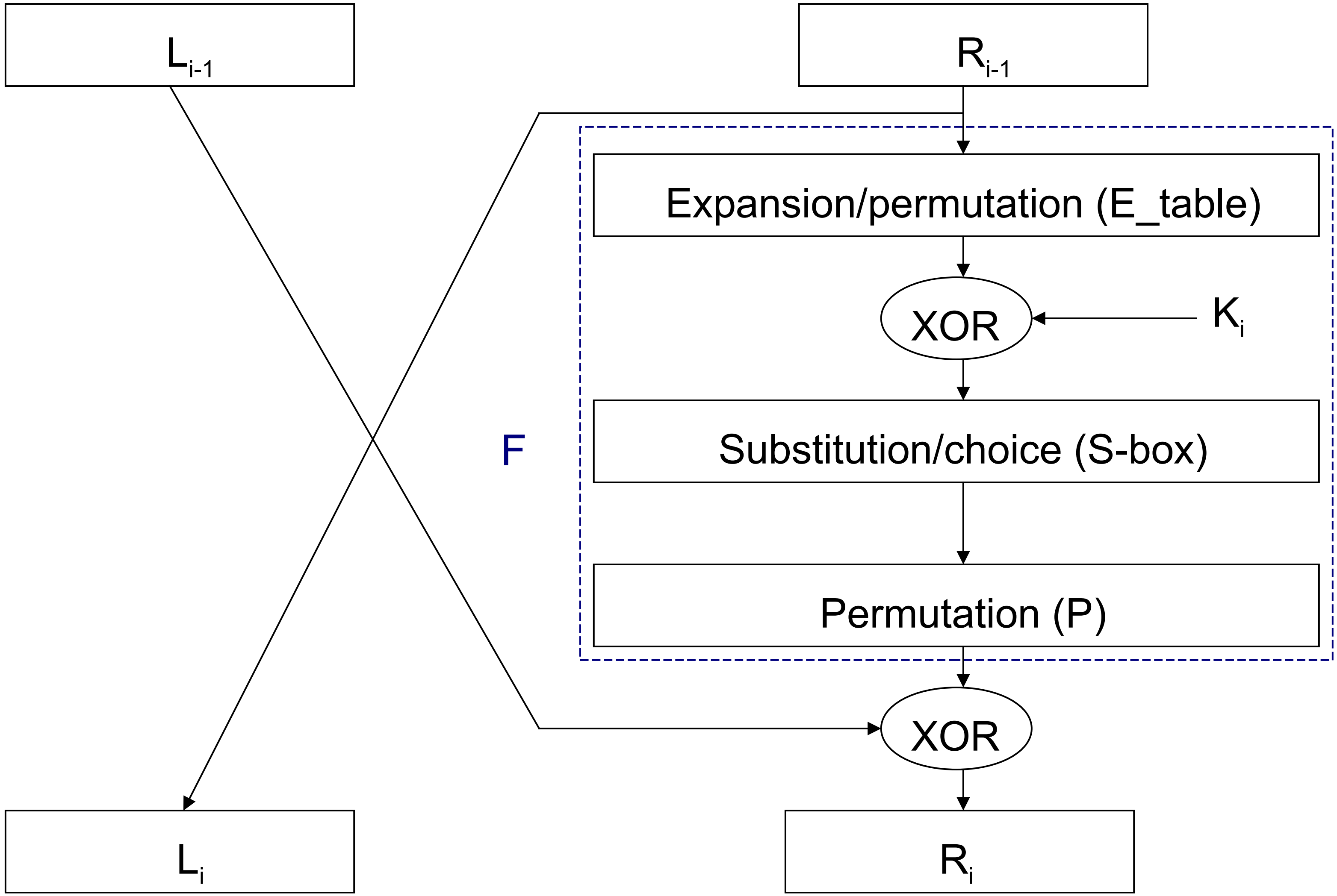


(Key Generation)



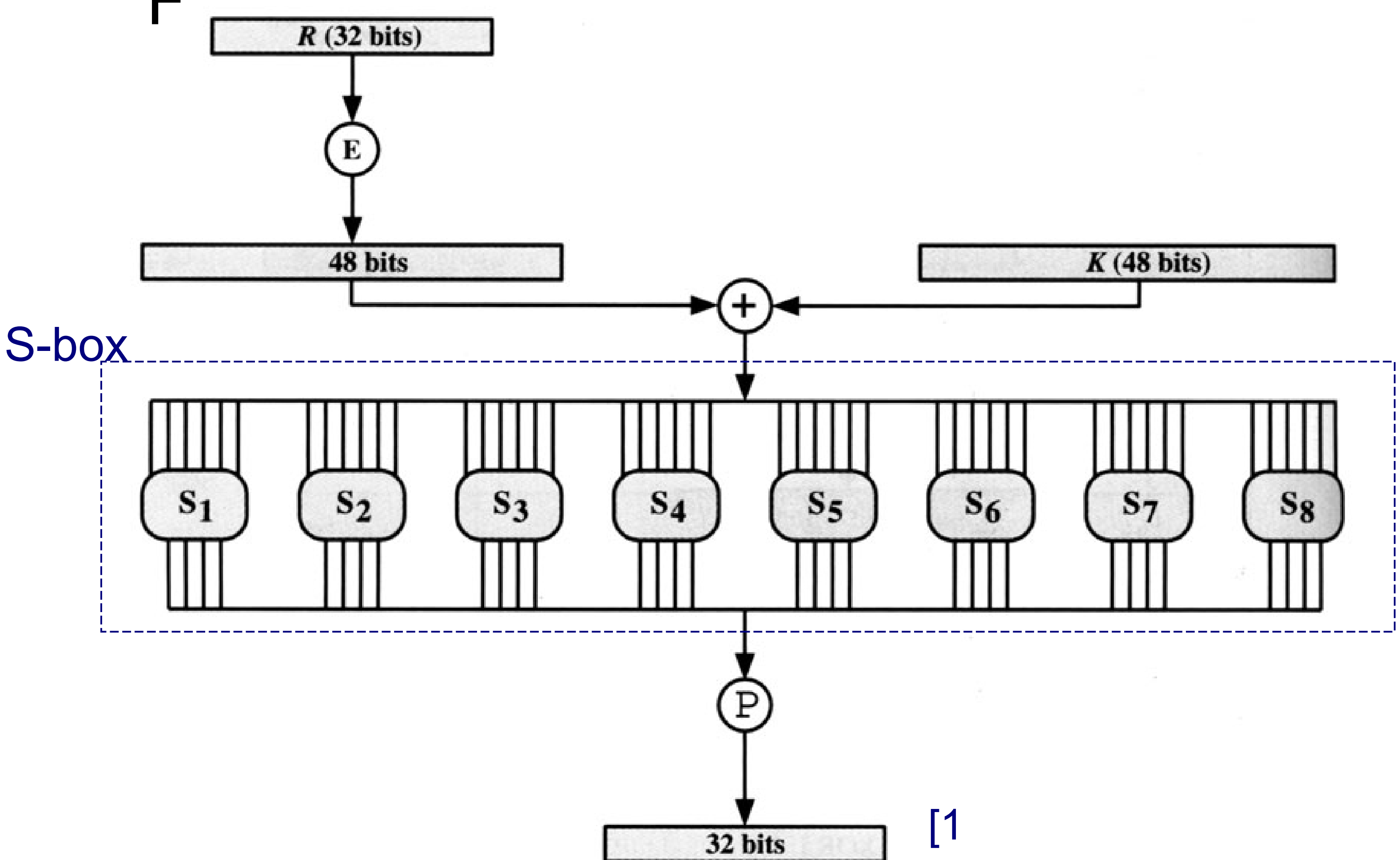
[1]

Encryption (Round) (cont.)



Encryption (Round) (cont.)

F



[1]

Encryption (Round) (cont.)

- Separate plaintext as L_0R_0
  - L_0: left half 32 bits of plaintext
  - R_0: right half 32 bits of plaintext
- Expansion/permutation: E( )
- Substitution/choice: S-box( )
- Permutation: P( )

F

-
- $R_i = L_{i-1} \sim P(S\_box(E(R_{i-1}) \sim Key_i))$

  $L_i = R_{i-1}$

Encryption (Round) (cont.)

- E

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	45	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

Expansion

Expansion

- 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
9	13	30	6	22	11	4	25

Encryption (Round) (cont.)

- S-box

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

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

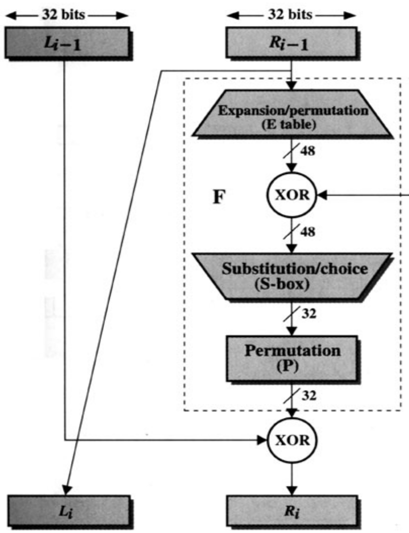
S_7	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

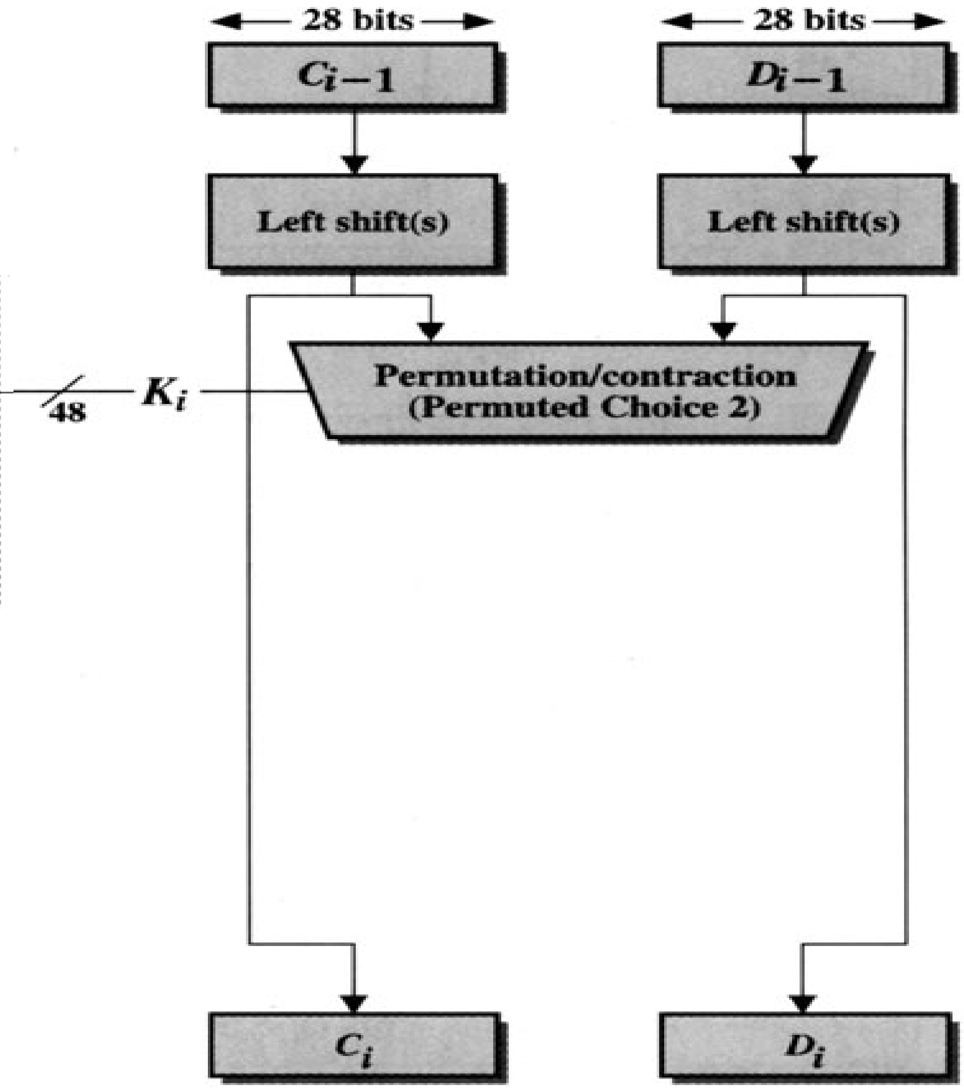
S_8	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

[1]

Key Generation

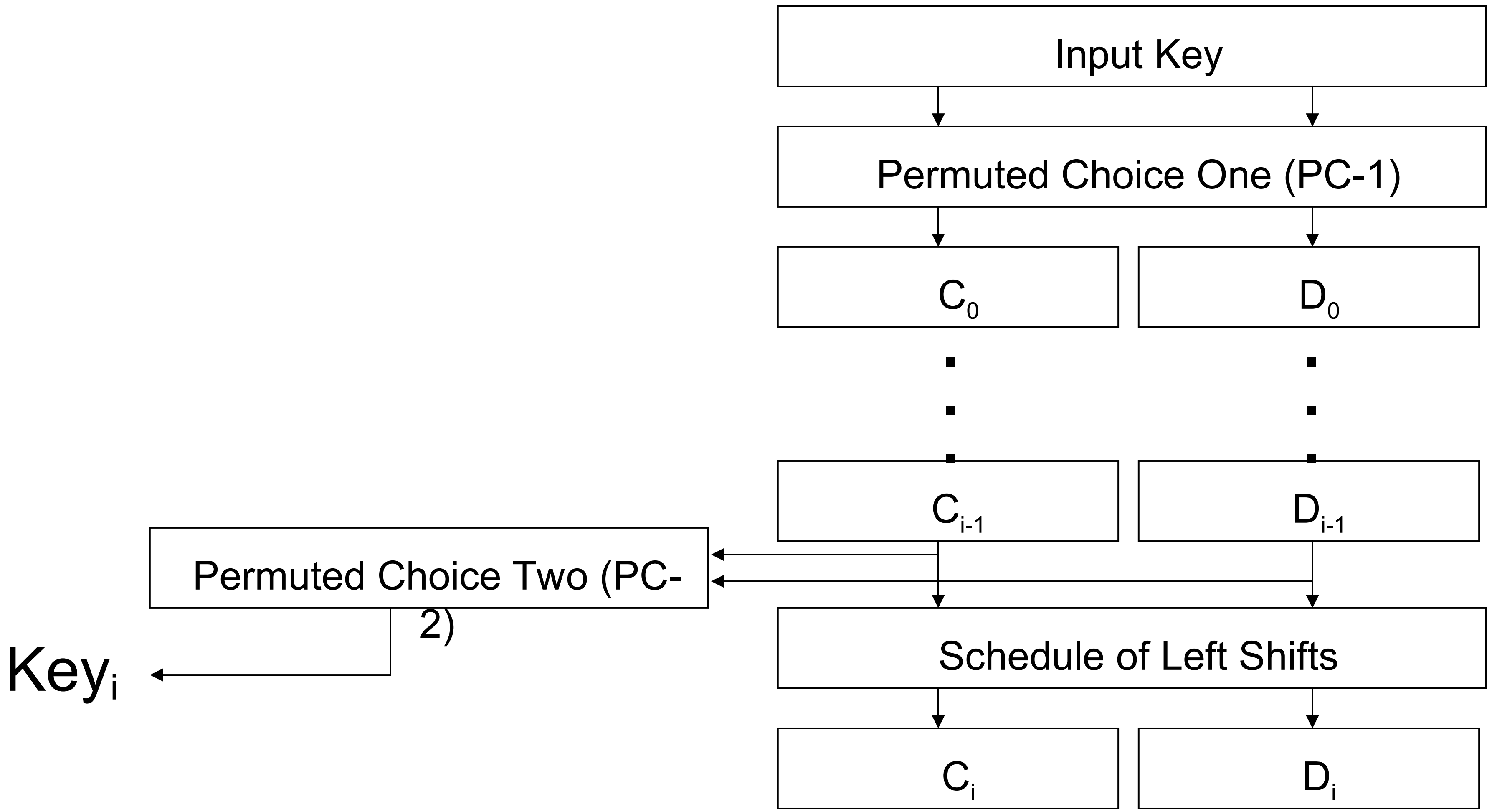
(Encryption)





[1]

Key Generation (cont.)



Permutation Choice 1

Left

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

Right

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

Permuted Choice 2

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

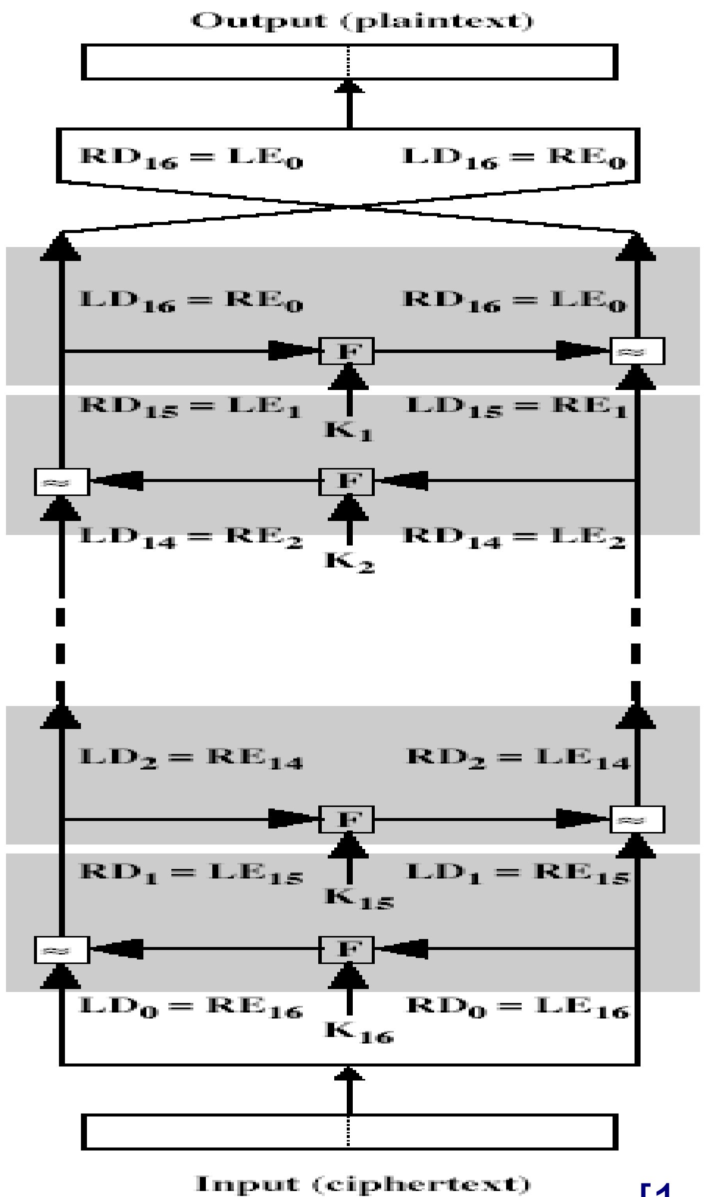
Key Generation (cont.)

- Original Key: Key_0
- Permuted Choice One: PC_1( )
- Permuted Choice Two: PC_2( )
- Schedule of Left Shift: SLS( )
- $(C_0, D_0) = PC\_1(Key_0)$
- $(C_i, D_i) = SLS(C_{i-1}, D_{i-1})$
- $Key_i = PC\_2(SLS(C_{i-1}, D_{i-1}))$

Expansion table (E-box)					
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

Decryption

- The same algorithm as encryption.
- Reversed the order of key (Key_{16}, Key_{15}, … Key_{1}).
- For example:
  - IP undoes IP^{-1} step of encryption.
  - 1st round with SK16 undoes 16th encrypt round.



[1]

Strength of DES

- Criticism
  - Reduction in key size of 72 bits
    - Too short to withstand with brute-force attack
  - S-boxes were classified.
    - Weak points enable NSA to decipher without key.
- 56-bit keys have $2^{56} = 7.2 \times 10^{16}$ values
  - Brute force search looks hard.
  - A machine performing one DES encryption per microsecond would take more than a thousand year to break the cipher.

Strength of DES (cont.)

- Avalanche effect in DES
  - If a small change in either the plaintext or the key, the ciphertext should change markedly.
- DES exhibits a strong avalanche effect.

(a) Change in Plaintext		(b) Change in Key	
Round	Number of bits that differ	Round	Number 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
9	42	9	40
10	44	10	38
11	32	11	31
12	30	12	33
13	30	13	28
14	26	14	26
15	29	15	34
16	34	16	35

Ultimate

- DES was proved insecure
  - In 1997 on Internet in a few months
  - in 1998 on dedicated h/w (EFF) in a few days
  - In 1999 above combined in 22hrs!

References

- [1] William Stallings, *Cryptography and Network Security*, 1999.