Lab Assign: 2

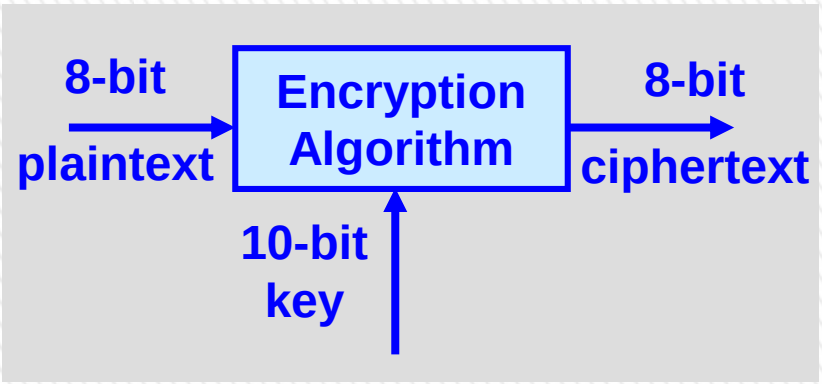
Write a program using JAVA or Python or C++ to implement Feistal Cipher structure

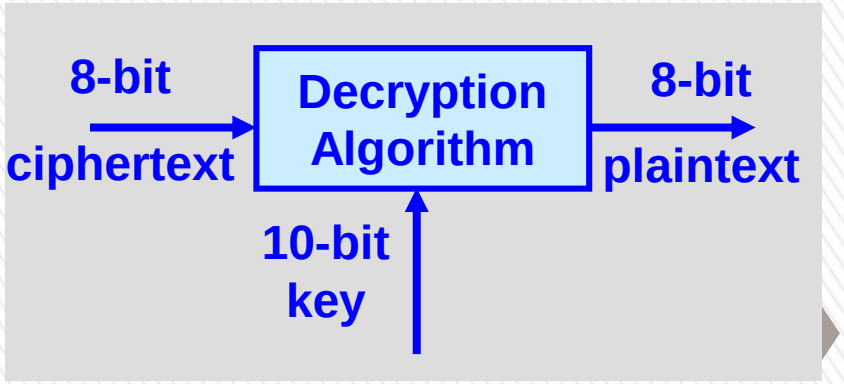
Simplified DES

- *Simplified DES* (*S-DES*) was developed by *Professor Edward Schaefer* of *Santa Clara University*.
- It is *an educational* rather than a *secure encryption algorithm*.
- It has *similar properties* and *structure* to *DES* with much *smaller parameters*.

Simplified DES (Overview)

- The *S-DES* encryption algorithm takes an *8-bit block of plaintext* and a *10-bit key* as *input* and produces an *8-bit block of ciphertext* as *output*.
- The *S-DES* decryption algorithm takes an *8-bit block of ciphertext* and the same *10-bit key* used to produce the ciphertext as *input* and produces the original *8-bit block of plaintext* as *output*.

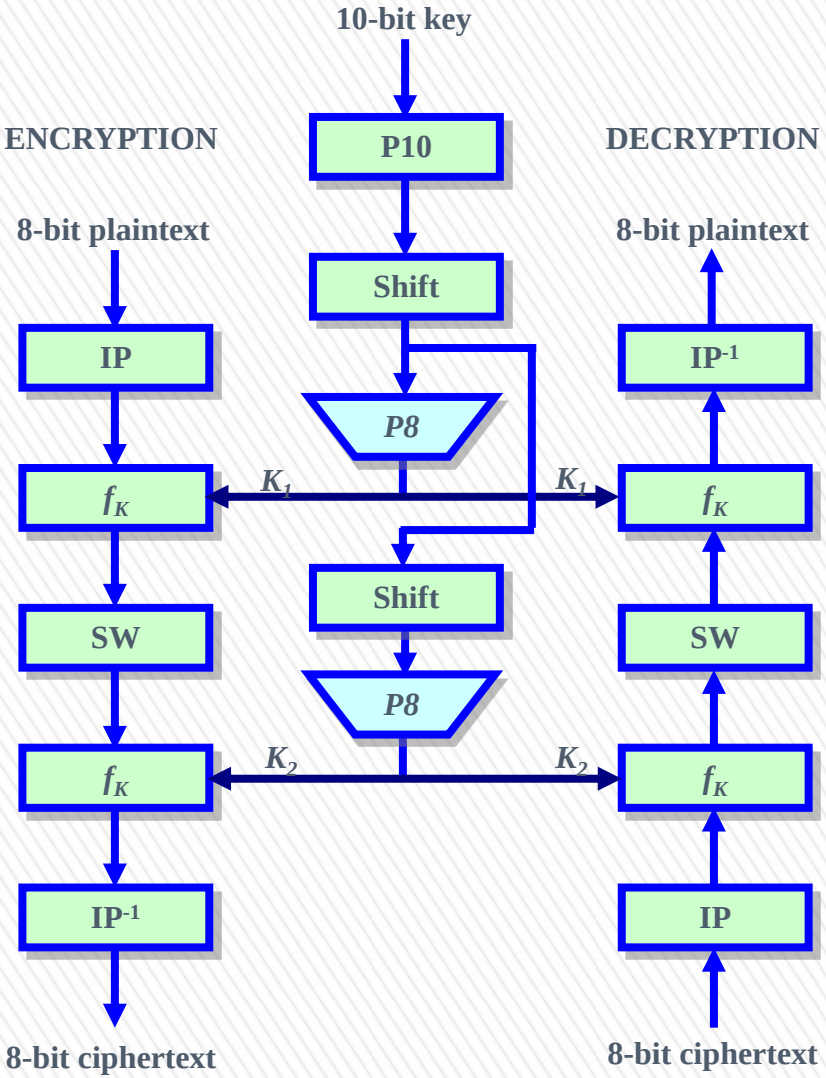




Simplified DES (STEPS)

- The *DES encryption algorithm* involves five functions:
  1. An *initial permutation* (*IP*);
  2. A *complex function* called f_k, which involves both *permutation* and *substitution* operations and depends on a *key input*;
  3. A *simple permutation function* that *switches* (*SW*) the two halves of the data;
  4. The function f_k again;
  5. A *permutation function* that is the *inverse* of the *initial permutation* (IP^{-1}).

Simplified DES Scheme



Plaintext:	1	1	1	1	0	0	1	1

Key:	1	0	1	0	0	0	0	0	1	0

IP:	2	6	3	1	4	8	5	7

IP^{-1}	4	1	3	5	7	2	8	6

$$S0 = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 1 & 0 & 3 & 2 \\ 3 & 2 & 1 & 0 \\ 0 & 2 & 1 & 3 \\ 3 & 1 & 3 & 2 \end{bmatrix} \end{array} \qquad S1 = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 0 & 1 & 2 & 3 \\ 2 & 0 & 1 & 3 \\ 3 & 0 & 1 & 0 \\ 2 & 1 & 0 & 3 \end{bmatrix} \end{array}$$

P-Boxes

P10

3	5	2	7	4	10	1	9	8	6

P8

6	3	7	4	8	5	10	9

P4

2	4	3	1
1	2	3	4

E/P

4	1	2	3	2	3	4	1

Simplified DES (Overview)

- The use of *multiple stages of permutation and substitution* results in a more complex algorithm, which increases the *difficulty of cryptanalysis*.
- The function f_k takes as input the data and 8-bit key.
- The algorithm can work with *16-bit key*, consisting of two *8-bit subkeys*, one used for each occurrence of f_k, or a *single 8-bit key* used twice in the algorithm.
- A compromise is to use a *10-bit key* from which two *8-bit subkeys* are *generated*.

Simplified DES (Mathematical Eqn)

- The *S-DES encryption algorithm* can be expressed as a composition of functions:

  $IP^{-1} \circ f_{K2} \circ SW \circ f_{K1} \circ IP$

- It can also be written as

  ciphertext = $\mathbf{IP^{-1}}(\boldsymbol{f_{K2}}\ (\mathbf{SW}(\boldsymbol{f_{K1}}(\mathbf{IP}(\text{plaintext})))))$

  K_1 = P8(Shift(P10(key)))

  K_2 = P8(Shift(Shift(P10))))

- The *S-DES decryption algorithm* is expressed as:

  plaintext = $IP^{-1}(f_{K1}\ (SW(f_{K2}(IP(\text{ciphertext})))))$

S-DES Key Generation

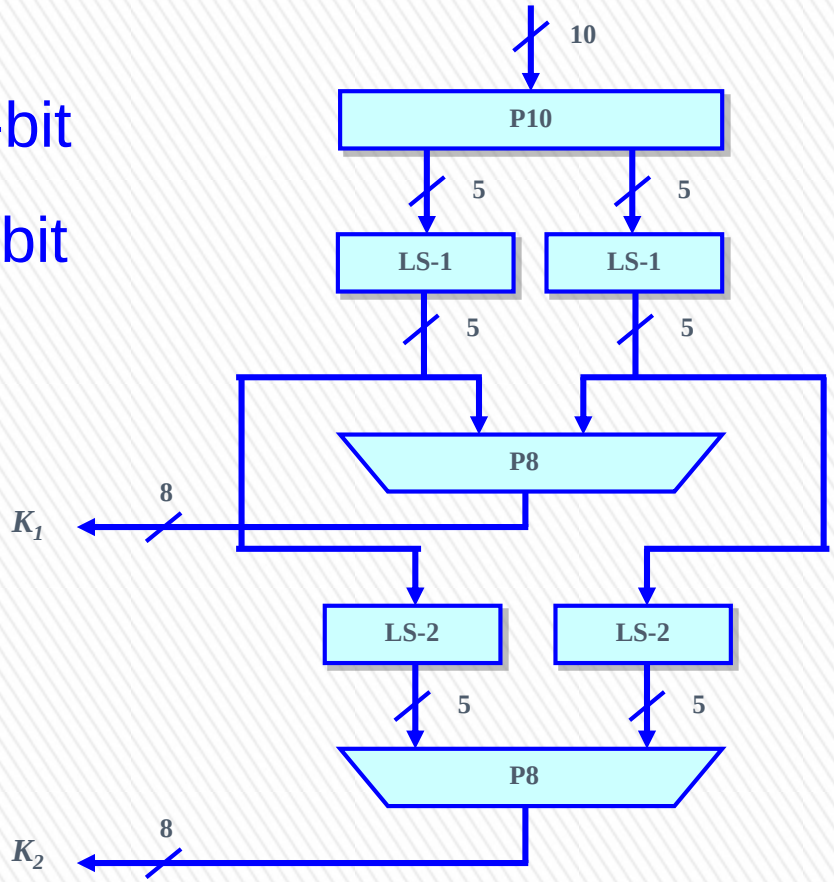
- *S-DES* depends on the use of a *10-bit key* shared between both sender and receiver.
- From this key, two *8-bit subkeys are generated* for use in stages of the encryption and decryption algorithms.

- Steps for Key generation
  - Initial permutation P10
  - Divide in left and right parts
  - Left shift and Merge
  - An 8 bits permutation, resulting in a 8 bits K1
  - Divide in left and right parts
  - Double left shift and Merge
  - An 8 bits permutation, resulting in a 8 bits K2

S-DES Key Generation

LS-1: Left-Shift 1-bit

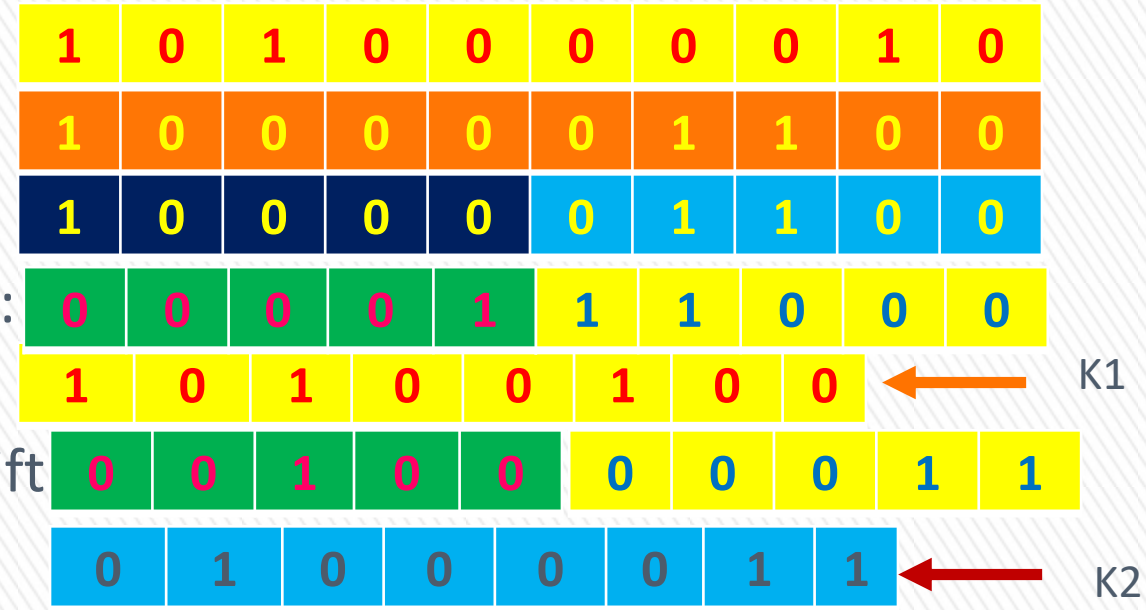
LS-2: Left Shift 2-bit



Key generation for simplified DES

Simplified S-DES (S-DES Key Generation)

» Example of key generation:



P10									
3	5	2	7	4	10	1	9	8	6

P8							
6	3	7	4	8	5	10	9

Simplified S-DES (S-DES Key Generation)

Worked Example for 10-bit input key (1010000010)

P10									
3	5	2	7	4	10	1	9	8	6

K_1	P8								K_2
10100100	6	3	7	4	8	5	10	9	01000011

S-DES Encryption

» *S-DES encryption* involves the *sequential application* of the *five functions* mentioned earlier.

Initial and Final Permutation

IP							
2	6	3	1	4	8	5	7

IP^{-1}							
4	1	3	5	7	2	8	6

$$IP^{-1}(IP(X)) = X$$

1	1	1	1	0	0	1	1

Simplified DES (S-DES Encryption)

The Function f_K

- The most complex component of *S-DES encryption* is the *function* f_K, which consists of *permutation* and *substitution functions*. The function is expressed as:

$$f_K(L, R) = (L \oplus F(R, SK), R)$$

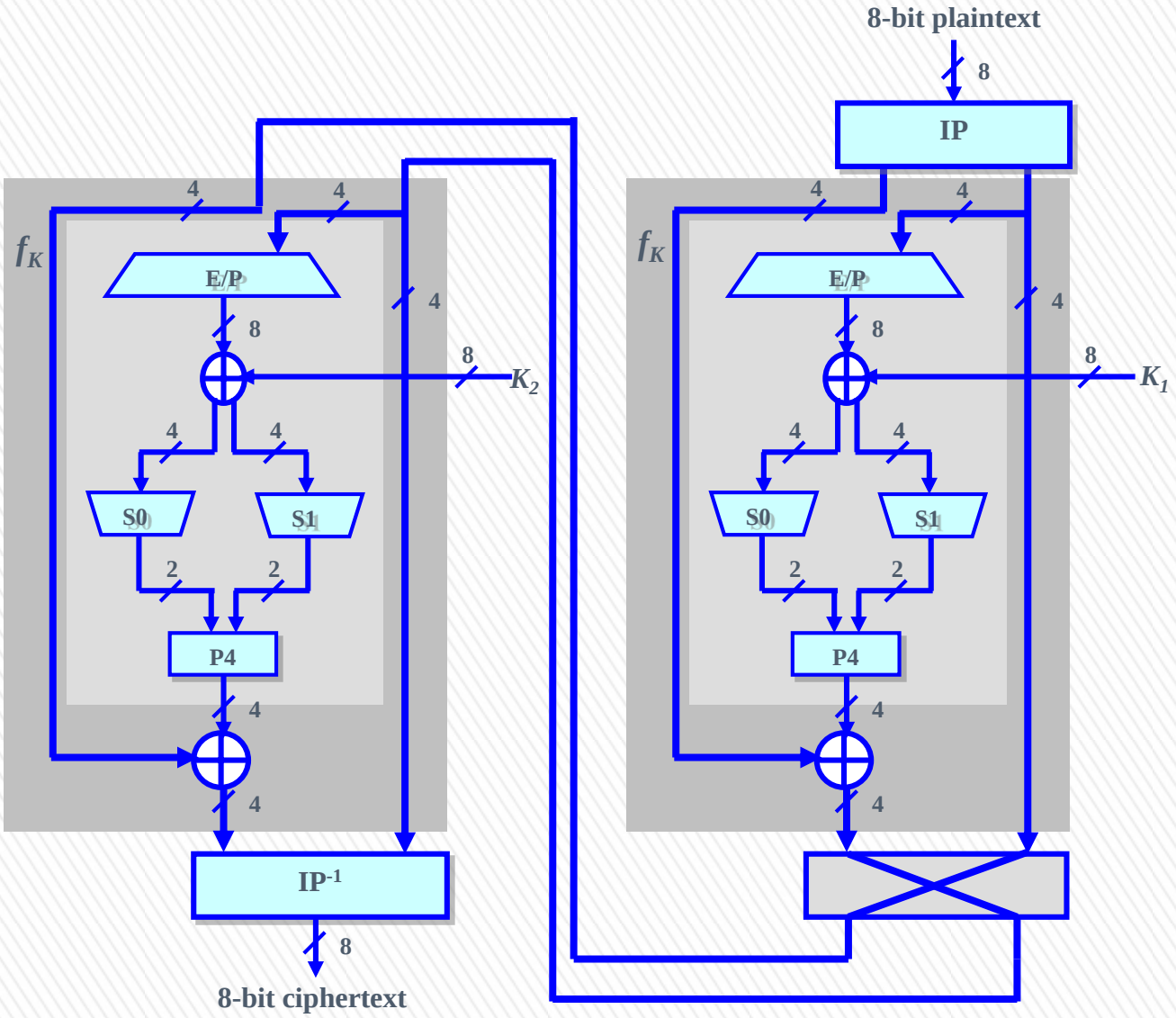
- *L* and *R* are the *leftmost 4 bits* and *rightmost 4 bits* of the *8-bit input* of f_K.
- *F* is a *mapping* from *4-bit strings* to *4-bit strings*.
- *SK* is a *subkey*.
- $\oplus$ is the *bit-by-bit exclusive OR function*.

Simplified DES (S-DES Encryption)

The Function *F*

- The function F is taken from S0 and S1, such as:
  - R is expanded by E
  - The expansion is xored with the subkey
  - The first 4 bits are the input for S0, the last are input to S1
  - If the input to Si is $I_1I_2I_3I_4$, then I_1I_4 is the row to consider and I_2I_3 is the column
  - The output goes then through P4

Simplified DES Scheme Encryption Detail



Plain text

1	1	1	1	0	0	1	1

Initial Permutation

IP							
2	6	3	1	4	8	5	7

Output of IP

1	0	1	1	1	1	0	1

Compute F function

Divide the IP o/p into two half's (L & R)

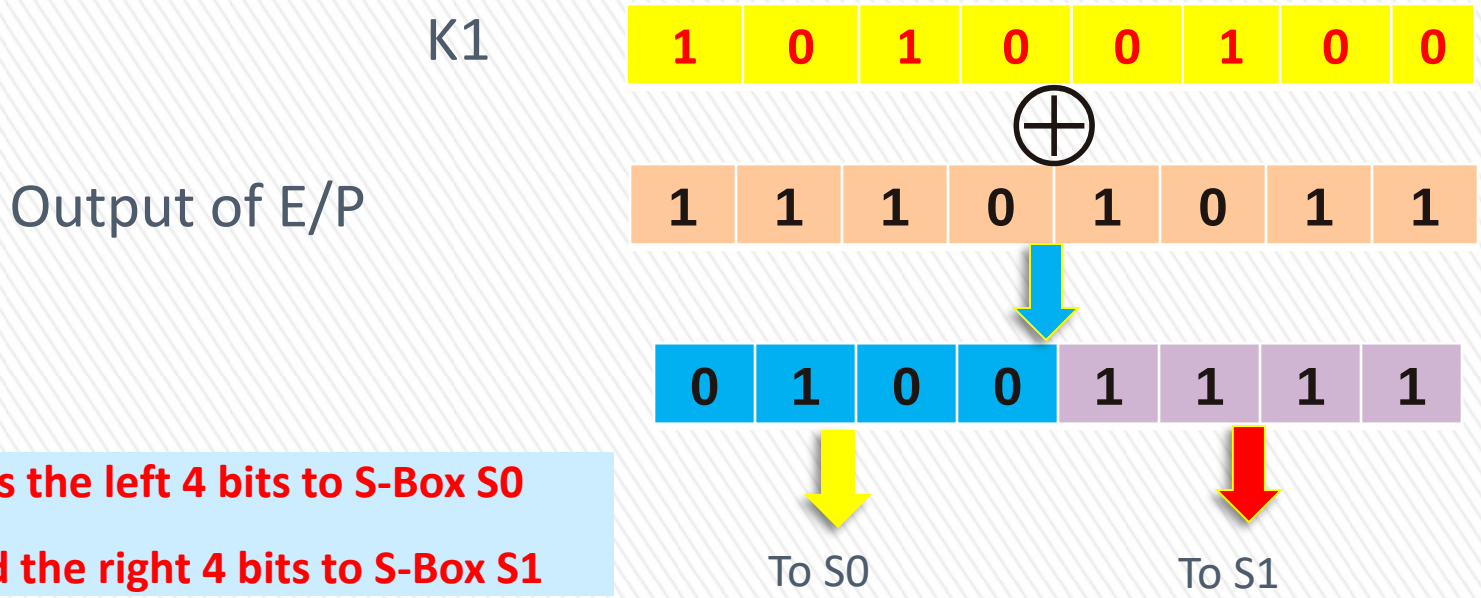
L				R			
1	0	1	1	1	1	0	1

Apply Expansion / Permutation E/ P to input 4 bits (R)

E/P							
4	1	2	3	2	3	4	1

1	1	1	0	1	0	1	1

Add the Output of E/P to sub key (k1) use (XOR)



Pass the left 4 bits to S-Box S0

And the right 4 bits to S-Box S1

$$S0 = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 1 & 0 & 3 & 2 \\ 3 & 2 & 1 & 0 \\ 0 & 2 & 1 & 3 \\ 3 & 1 & 3 & 2 \end{bmatrix} \end{array}$$

$$S1 = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 0 & 1 & 2 & 3 \\ 2 & 0 & 1 & 3 \\ 3 & 0 & 1 & 0 \\ 2 & 1 & 0 & 3 \end{bmatrix} \end{array}$$

S- Box Operation

1. First and fourth bits give row number
2. Second and third give column number
3. Look-up number in specified row and column
4. Covert to Binary

For L which is the input to S0

0	1	0	0

Row=00 (0) col=10 (2) → the output of S0=3 (11)

For R which is the input to S1

1	1	1	1

Row=11 (3) col=11 (3) → the output of S1=3 (11)

The output of S-Box is

1	1	1	1

Apply Permutation P4

P4			
2	4	3	1

The output of F Function

1	1	1	1

The Function f_K

$$f_K(L,R) = (L \oplus F(R,SK), R)$$

The output of F Function

1	1	1	1

L (The leftmost 4-bits of IP Output)

1	0	1	1

L Xor output of F

0	1	0	0

R (The Rightmost 4-bits of IP Output)

1	1	0	1

The Output of Function f_K

0	1	0	0	1	1	0	1

Simplified DES (S-DES Encryption)

The Switch Function

- The function f_K only *alters* the *leftmost 4 bits* of the input.
- The *switch function* (*SW*) *interchanges* the *left* and *right 4 bits* so that the second instance of f_K *operates on* a *different 4 bits*.
- In the second instance, the *E/P*, *S0*, *S1*, and *P4 functions are the same*.
- The *key* input is K_2.

Simplified DES (S-DES Encryption)

- f_{K1} (1011 1101) = ($L \oplus F(R, K_1)$, R)
  = (1011⊕1111,1101) = 0100 1101
- SW (0100 1101)= 1101 0100 = L || R
- F(R, K_2)
  - E/P (0100) ⊕ K_2= 00101000 ⊕ 01000011 = 01101011
  - S0 (0110) = 10
  - S1 (1011) = 01
  - P4 (1001) = 0101
- f_{K2}(1101 0100) = ($L \oplus F(R, K_2)$, R)
  = (1101⊕0101, 0100) = 1000 0100
- IP^{-1} (10000100) = 01000001

» Ciphertext C=01000001

Simplified DES (S-DES Decryption)

» $C = IP^{-1} \bullet f_{K2} \bullet SW \bullet f_{K1} \bullet IP\ (P)$

» $IP^{-1} \bullet f_{K1} \bullet SW \bullet f_{K2} \bullet IP\ (C)$

$= IP^{-1} \bullet f_{K1} \bullet SW \bullet f_{K2} \bullet IP \bullet IP^{-1} \bullet f_{K2} \bullet SW \bullet f_{K1} \bullet IP\ (P)$

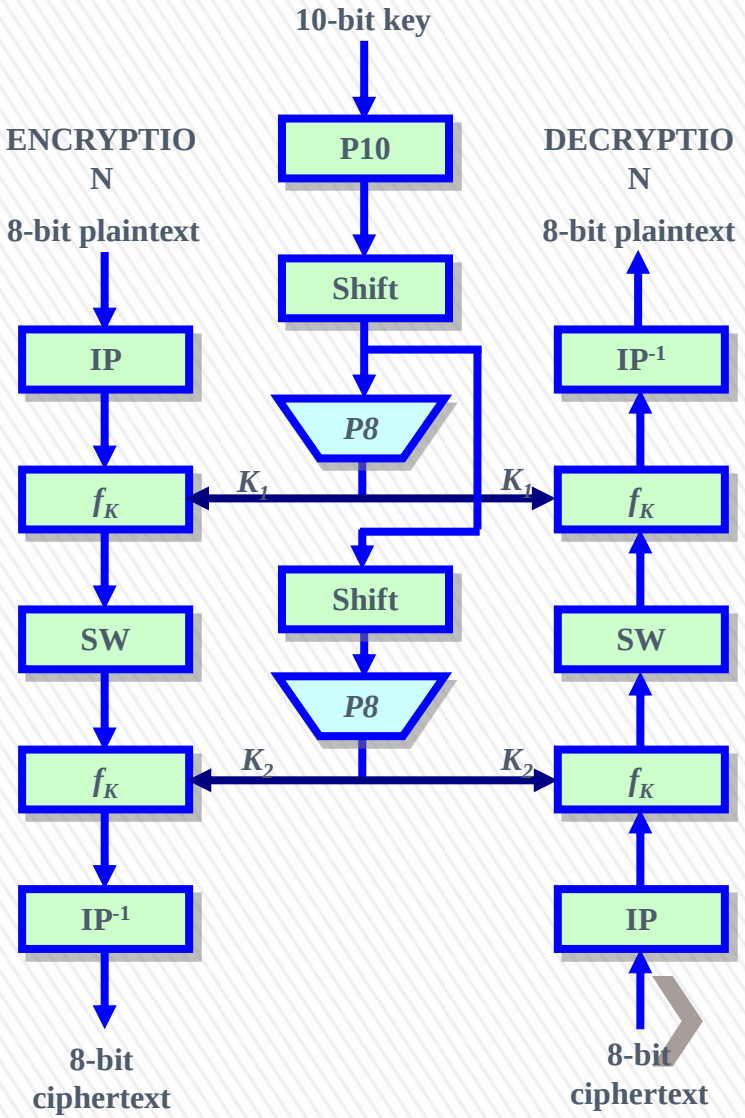
$= IP^{-1} \bullet f_{K1} \bullet SW \bullet f_{K2} \bullet f_{K2} \bullet SW \bullet f_{K1} \bullet IP\ (P)$

$= IP^{-1} \bullet f_{K1} \bullet SW \bullet SW \bullet f_{K1} \bullet IP\ (P)$

$= IP^{-1} \bullet f_{K1} \bullet f_{K1} \bullet IP\ (P)$

$= IP^{-1} \bullet IP\ (P)$

$= P$



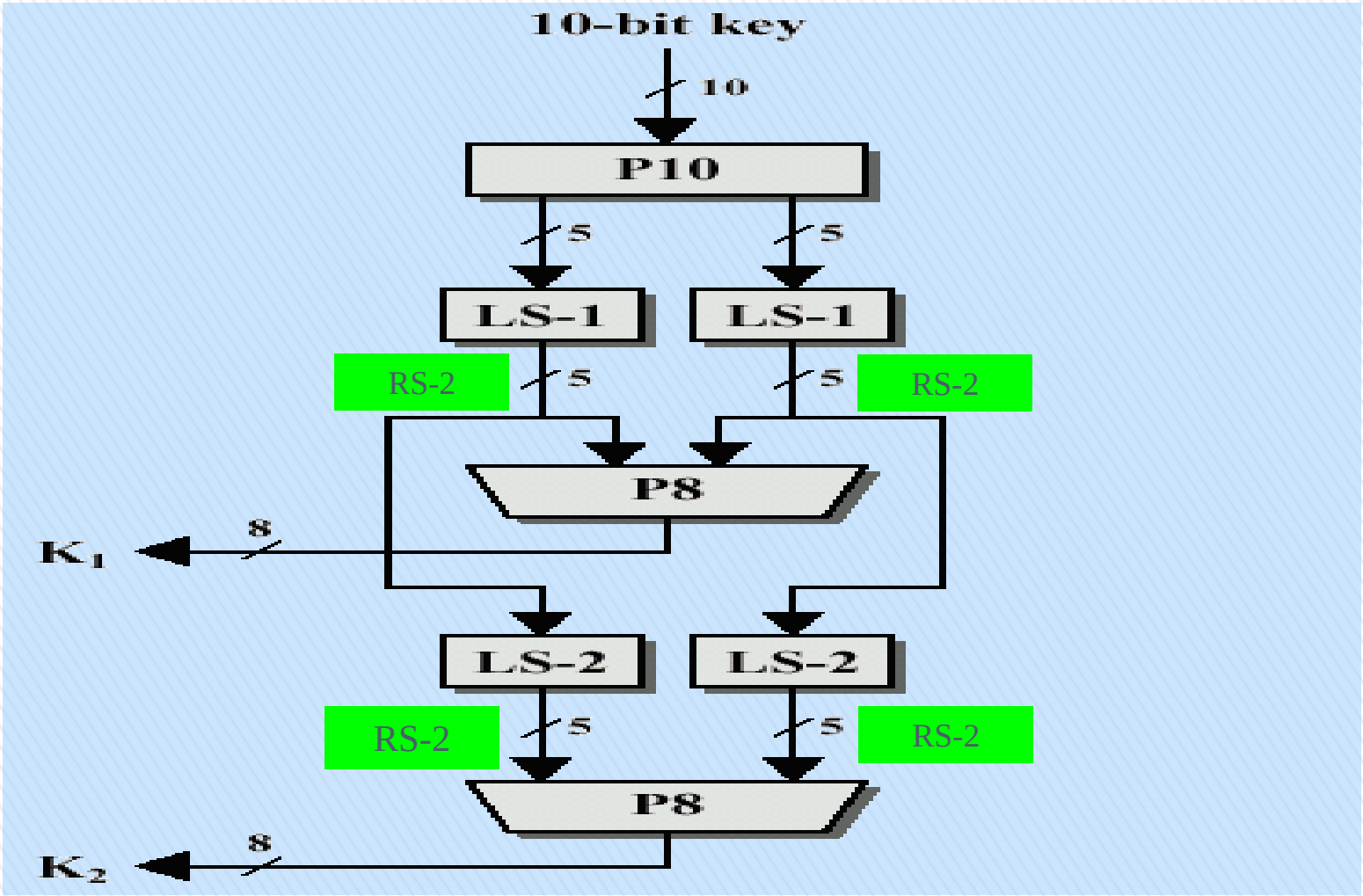
Simplified DES (S-DES Decryption)

- Only **sub-keys are fed in reverse order**
- SW • SW = I (identity)
- $IP^{-1} \bullet IP = IP \bullet IP^{-1} = I$ (identity)
- $f_{K1} \bullet f_{K1}(X,Y) = f_{K1}(X \oplus F(Y, K_1), Y)$
  $= (X \oplus F(Y, K_1) \oplus F(Y, K_1), Y)$
  $= (X, Y)$
- $f_{K2} \bullet f_{K2}(X,Y) = f_{K2}(X \oplus F(Y, K_2), Y)$
  $= (X \oplus F(Y, K_2) \oplus F(Y, K_2), Y)$
  $= (X, Y)$

Simplified DES (S-DES Decryption)

- Generate sub-keys in reverse order
- P10(K)=k1 k2 … k10
- Encryption
  - LS-1(k1 k2 k3 k4 k5) = k2 k3 k4 k5 k1
  - LS-2 (k2 k3 k4 k5 k1) = k4 k5 k1 k2 k3
- Decryption
  - RS-2 (k1 k2 k3 k4 k5) = k4 k5 k1 k2 k3
  - RS-2 (k4 k5 k1 k2 k3) = k2 k3 k4 k5 k1

Generate sub-keys in reverse order



Simplified DES (Analysis of S-DES)

- A *brute-force attack* on *S-DES* is certainly feasible, since for *10-bit key*, there are only *1024 possibilities*.
- Given a *ciphertext*, an attacker can try each possibility and analyse the result to determine if it is a *reasonable plaintext*.

Simplified DES (Analysis of S-DES)

- Cryptanalysis attack can be performed in two different ways:
  1. Derive *8 nonlinear equations* with *10 unknowns*. There are a number of solutions, but each of these could be calculated and then analysed.
  2. Each of the permutations and additions in the algorithm is a linear mapping. The *nonlinearity* comes from the *S-boxes*. Alternating linear maps with the S-boxes nonlinear maps results in *very complex polynomial expressions* for the *ciphertext bits*, making *cryptanalysis very difficult*.

Simplified DES (Relationship to DES)

- *DES* operates on *64-bit blocks* of input.
- The *encryption scheme* can be defined as:

  $IP^{-1} \circ f_{K16} \circ SW \circ f_{K15} \circ SW \circ \ldots \circ SW \circ f_{K1} \circ IP$
- A *56-bit key* is used, from which *sixteen 48-bit subkeys* are calculated.

Simplified DES (Relationship to DES)

- The *sequence of operations* are as follows:
  - *Initial permutation* of *56-bit* followed by a sequence of *shifts* and *permutations* of *48 bits*.
  - Within the *encryption algorithm*, instead of *F* acting on *4 bits* (n_1, n_2, n_3, n_4), it acts on *32 bits* ($n_1, \ldots, n_{32}$).
  - After the *initial E/P*, the output of *48 bits* can be diagrammed as:

n_{32}	n_1	n_2	n_3	n_4	n_5
n_4	n_5	n_6	n_7	n_8	n_9
•	•	•	•	•	•
•	•	•	•	•	•
•	•	•	•	•	•
n_{28}	n_{29}	n_{30}	n_{31}	n_{32}	n_1

Simplified DES (Relationship to DES)

- This matrix is added (*XOR*) to a *48-bit subkey*.
- There are *8 rows* corresponding to *8 S-boxes*.
- Each *S-box* has *4 rows* and *16 columns*.
- The *first* and the *last bit* of a *row* of the preceding matrix *picks out a row* of an *S-box*, and the *middle 4 bits pick out a column*.