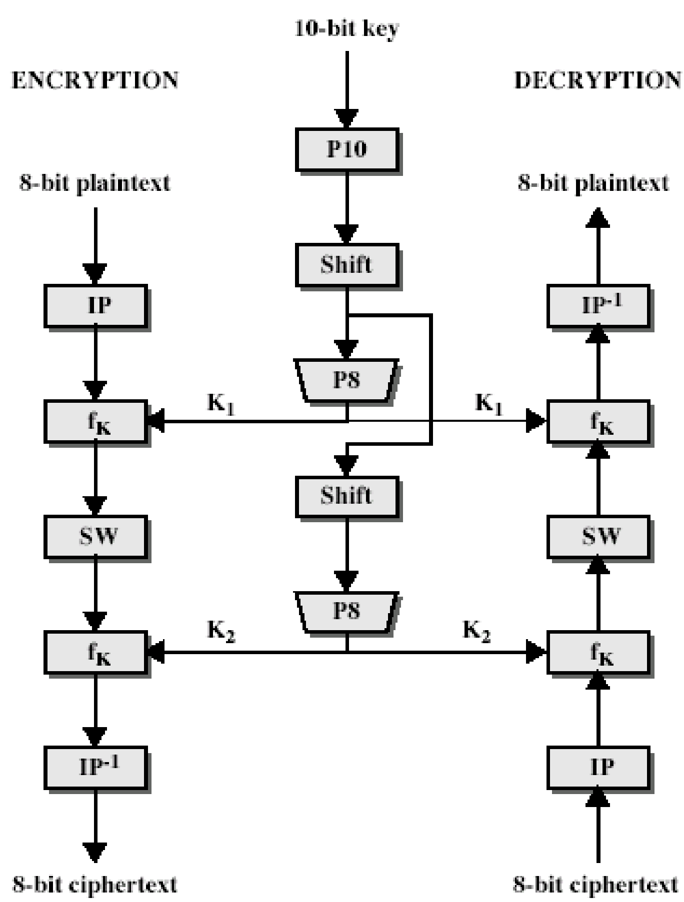
3 CONVENTIONAL ENCRIPTION: MODERN TECHNIQUES

We will focus on the most widely used conventional encryption algorithms: the Data Encryption Standard (DES). Although numerous conventional encryption algorithms have been developed since the introduction of DES, it remains the most important such algorithm.

Simplified DES

The S-DES encryption algorithm takes an 8-bit block of plaintext (example: 10111101) 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 that ciphertext as input and produces the original 8-bit block of plaintext.



Simplified DES Scheme

The encryption algorithm involves five functions: an initial permutation (IP); a complex function labvelled as f_k, which involves both permutation and substitution operations and depends on on a key input; a simple permutation function that switches (SW) the two halves of the data; the function f_k again, and finally a permutation function that is the inverse of the initial permutation (IP^{-1}). 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 not only the data passing through the encryption algorithm, but also an 8-bit key. The algorithm could have been designed to work with a 16-bit key, consisting of two 8-bit subkeys, one used for each occurrence of f_k. Alternatively, a single 8-bit key could have been used, with the same key used twice in the algorithm. A compromise is to use a 10-bit key from which two 8-bit subkeys are generated as depicted in the figure below. In this case, the key is first subjected to permutation (P10). Then a shift operation is performed. The output of shift operation then passes through a permutation function that produces an 8-bit output (P8) for the first subkey (K_1). The output of the shift operation also feeds into another shift and another instance of P8 to produce the second subkey (K_2).

S-DES key generation. S-DES depends on the use of a 10-bit key shared between sender and receiver. From this key, two 8-bit subkeys are produces for use in particular stages of the encryption and decryption algorithm. The stages to produce the keys are illustrated below.
First, permute the key in the following fashion. Let the 10-bit key be designed as (k_1 , k_2 , k_3 , k_4 , k_5 , k_6 , k_7 , k_8 , k_9 , k_{10}). Then the permutation P10 is defined as

P10 (k_1 , k_2 , k_3 , k_4 , k_5 , k_6 , k_7 , k_8 , k_9 , k_{10}) = (k_3 , k_5 , k_2 , k_7 , k_4 , k_{10} , k_1 , k_9 , k_8 , k_6).

For example, the key (1010000010) is permuted to (1000001100). Next, perform a circular left shift (LS-1), or rotation, separately on the first five bits and the second five bits. In out example, the result is (00001 11000).

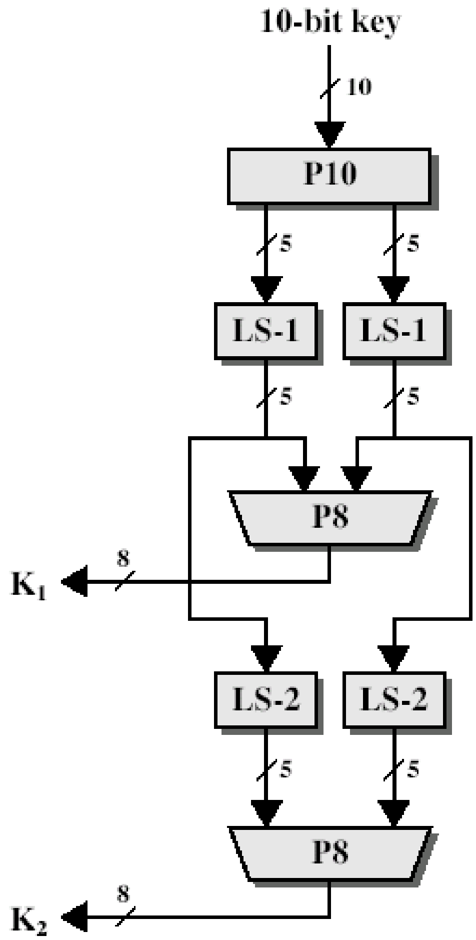
Next we apply P8, which picks out and permutes 8 of the 10 bits according to the following rule:

P8 : (6 3 7 4 8 5 10 9).

The result is subkey1(K_1). In our example, this yields (10100100).

We then go back to the pair of 5-bit strings produced by the two LS-1 functions and perform a circular left shift of 2 bit positions on each string. In our example, the value (00001

11000) becomes (00100 00011). Finally, P8 is applied again to produce K_2. In our example, the result is (01000011).



Key Generation for Simplified DES

S-DES encryption. As was mentioned, encryption involves the sequential application of five functions. We examine each of these.

Initial and final permutations. The input to the algorithm is an 8-bit block of plaintext, which we first permute using the IP function

IP : 26314857

This retains all 8 bits of the plaintext but mixes them up. At the end of the algorithm, the inverse permutation is used

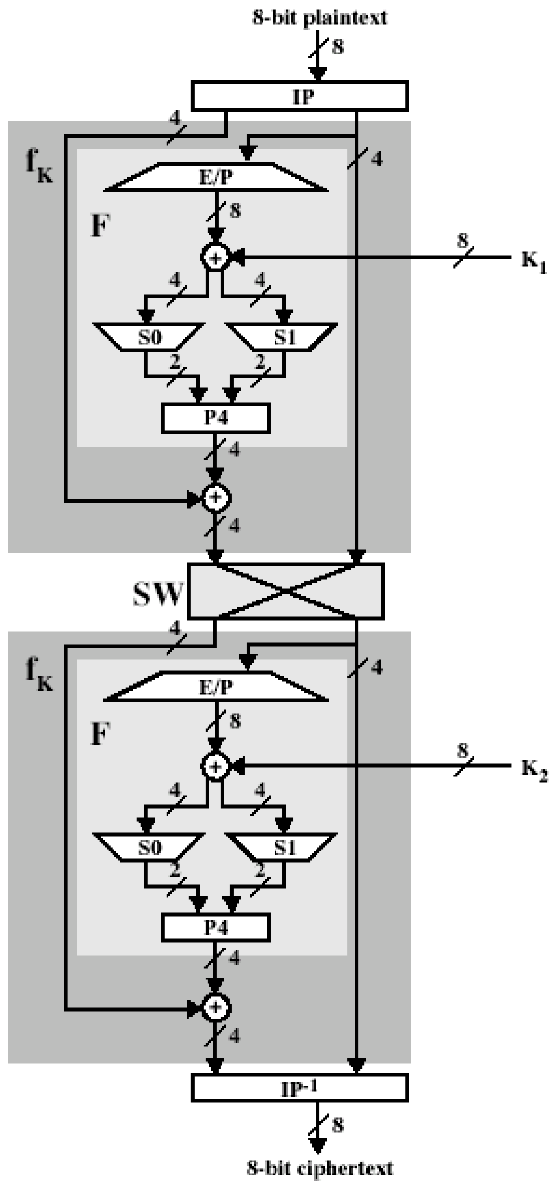
IP^{-1} : 41357286

Indeed, the second permutation is reverse of the first.

The function f_k. The most complex component of S-DES is the function f_k, which consists of a combination of permutation and substitution functions. The functions can be expressed as follows. Let *L* and *R* be the leftmost 4 bits and rightmost 4 bits of the 8-bit input to f_k, and let *F* be a mapping (not necessarily one-to-one) from 4-bit strings to 4-bit strings. Then we let

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

where *SK* is a subkey and $\oplus$ is the bit-by-bit exclusive-OR operation function.



Simplified DES Encryption Detail

For example, suppose the output of the IP stage is (10111101) and F(1101, SK) = (1110) for some key SK. Then f_k (10111101) = (01011101) because (1011) ⊕ (1110) = (0101).

We now describe the mapping F. The input is a 4-bit number (n_1 n_2 n_3 n_4). The first operation is an expansion/permutation operation

E/P : 41232341

For what follows, it is clearer to depict the results in this fashion:

$$\begin{array}{c|cc|c} n_4 & n_1 & n_2 & n_3 \\ n_2 & n_3 & n_4 & n_1 \end{array}$$

The 8-bit subkey $K_1 = (k_{11}, k_{12}, k_{13}, k_{14}, k_{15}, k_{16}, k_{17}, k_{18})$ is added to this value using exclusive-OR:

$$\begin{array}{c|cc|c} n_4 + k_{11} & n_1 + k_{12} & n_2 + k_{13} & n_3 + k_{14} \\ n_2 + k_{15} & n_3 + k_{16} & n_4 + k_{17} & n_1 + k_{18} \end{array}$$

Let us rename these bits:

$$\begin{array}{c|cc|c} p_{0,0} & p_{0,1} & p_{0,2} & p_{0,3} \\ p_{1,0} & p_{1,1} & p_{1,2} & p_{1,3} \end{array}$$

The first four bits (first row of the precedence matrix) are fed into the S-box S0 to produce a 2-bit output, and the remaining 4 bits (second row) are fed into S1 to produce another 2-bit output. These two are defined as follows:

$$S0 = \begin{pmatrix} 1 & 0 & 3 & 2 \\ 3 & 2 & 1 & 0 \\ 0 & 2 & 1 & 3 \\ 3 & 1 & 3 & 2 \end{pmatrix} \quad S1 = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 2 & 0 & 1 & 3 \\ 3 & 0 & 1 & 0 \\ 2 & 1 & 0 & 3 \end{pmatrix}$$

The S-boxes operate as follows: The first and fourth input bits are treated as 2-bit numbers and specify a row of the S-box, and the second the third input bits specify a column of the S-box. The entry in that row and column, in base 2, is the 2-bit output. For example, if $(p_{0,0}p_{0,3}) = (00)$ and $(p_{0,1}p_{0,2}) = (10)$, then the output is from row 0, column 2 of S0, which is 3, or (11) in binary. Similarly, $(p_{1,0}p_{1,3})$ and $(p_{1,1}p_{1,2})$ are used to index into a row and column of S1 to produce an additional 2 bits.

Next, the 4 bits produced by S0 and S1 undergo a further permutation as follows:

P4 : 2431

The output of P4 is the output of the function F.

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.

Example

S-DES Key Generation

10-bit key : 1100101001

Action	Input	Output
P10	1100101001	0111011000
LS-1	0111011000	1110010001
P8	1110010001	11000010 (K_1)
LS-2	1110010001	1001100110
P8	1001100110	00011101 (K_2)

S-DES Encryption

8-bit plaintext : 10100110

IP	10100110	01110001
E/P	0001	10000010
Excusive-OR	10000010, K_1	0100 0000
S0	0100	11
S1	0000	00
P4	1100	1001
Excusive-OR	0111, 1001	1110
SW	11100001	00011110
E/P	1110	01111101
Excusive-OR	01111101, K_2	01100000
S0	0110	10
S1	0000	00
P4	1000	0001
Excusive-OR	0001, 0001	0000
IP^{-1}	00001110	00011001

8-bit ciphertext : 00011001

S-DES Decryption

8-bit ciphertext : 00011001

IP	00011001	00001110
E/P	1110	01111101
Excusive-OR	01111101, K_2	01100000
S0	0110	10
S1	0000	00
P4	1000	0001
Excusive-OR	0000, 0001	0001
SW	00011110	11100001

E/P	0001	10000010
Excusive-OR	10000010, K_1	01000000
S0	0100	11
S1	0000	00
P4	1100	1001
Excusive-OR	1110, 1001	0111
IP^{-1}	01110001	10100110

8-bit plaintext : 10100110