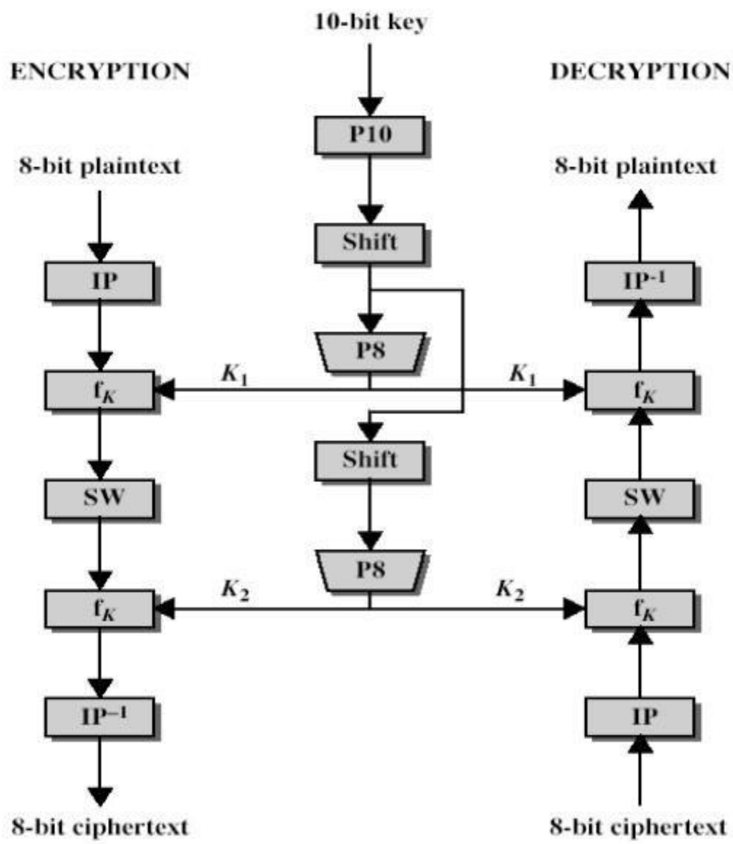
SIMPLIFIED DATA ENCRYPTION STANDARD (S-DES)

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

OVERVIEW

- 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.
- 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.



Reference :William Stallings, Cryptography and Network Security: Principles and Practice, PHI 3rd Edition, 2006

- The encryption algorithm involves five functions: an initial permutation (IP); a complex function labeled f_K, which involves both permutation and substitution operations and depends 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. In this case, the key is first subjected to a permutation (P10). Then a shift operation is

performed. The output of the 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).

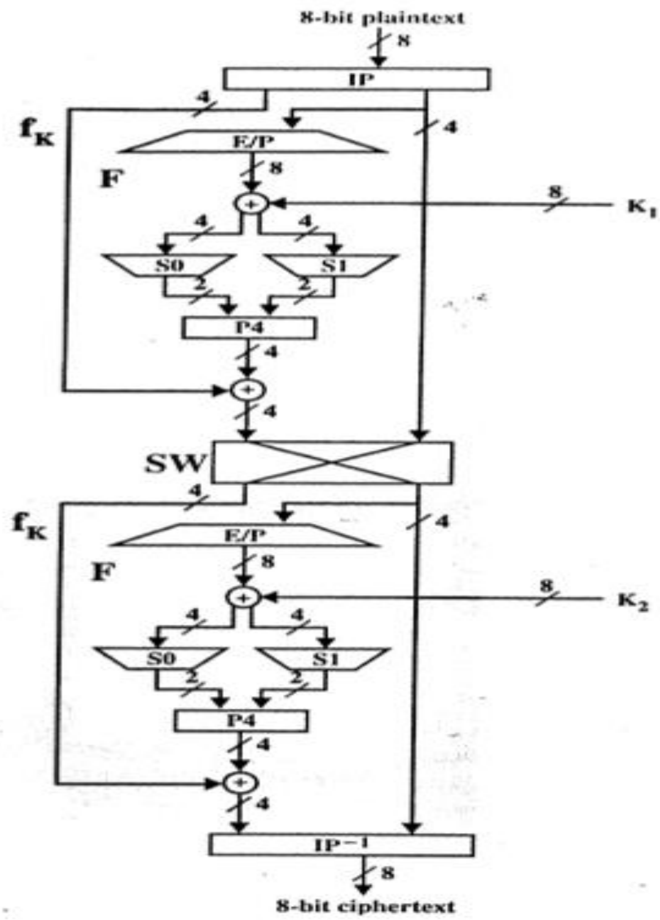


Figure 3.3 Simplified DES Scheme Encryption Detail.

Reference :William Stallings, Cryptography and Network Security: Principles and Practice, PHI 3rd Edition, 2006

- We can concisely express the encryption algorithm as a composition1 of functions:
    - IP-1 ○ fK2 ○ SW ○ fK1 ○ IP
- which can also be written as:
    - ciphertext = IP-1(fK2 (SW (fK1 (IP(plaintext)))))
- where K1 = P8(Shift(P10(key)))
    - K2 = P8(Shift(Shift(P10(key))))

- Decryption is essentially the reverse of encryption:
    - plaintext = IP-1(fK1 (SW (fK2 (IP(ciphertext)))))

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 produced for use in particular stages of the encryption and decryption algorithm

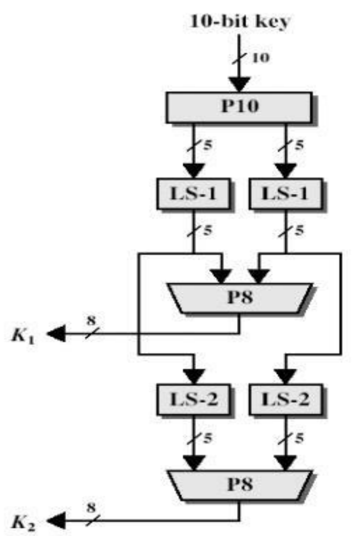


Figure: key generation for S-DES

Reference :William Stallings, Cryptography and Network Security: Principles and Practice, PHI 3rd Edition, 2006

- First, permute the key in the following fashion. Let the 10-bit key be designated as (k1, k2, k3, k4, k5, k6, k7, k8, k9, k10). Then the permutation P10 is defined as:
- P10(k1, k2, k3, k4, k5, k6, k7, k8, k9, k10) = (k3, k5, k2, k7, k4, k10, k1, k9, k8, k6)
- P10 can be concisely defined by the display:

P10									
3	5	2	7	4	10	1	9	8	6

- This table is read from left to right; each position in the table gives the identity of the input bit that produces the output bit in that position.
- So the first output bit is bit 3 of the input; the second output bit is bit 5 of the input, and so on. 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 our 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 subkey 1 (K1).
- 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 K2. In our example, the result is (01000011).

S-DES ENCRYPTION

- encryption involves the sequential application of five functions.
- 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 2 6 3 1 4 8 5 7
- 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 4 1 3 5 7 2 8 6
- It is easy to show by example that the second permutation is indeed the reverse of the first; that is, IP–1(IP(X)) = X.

The Function fK

- The most complex component of S-DES is the function fK, 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 fK, 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 ! F(R, SK), R)
- where SK is a subkey and ! is the bit-by-bit exclusive-OR function.
- For example, suppose the output of the IP stage is (10111101) and F(1101, SK) = (1110) for some key SK.
    - Then fK(10111101) = (01011101) because (1011) ! (1110) = (0101).
- We now describe the mapping F. The input is a 4-bit number (n1n2n3n4).
- The first operation is an expansion/permutation operation:

E/P							
4	1	2	3	2	3	4	1

- For what follows, it is clearer to depict the result 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 K1 = (k11, k12, k13, k14, k15, k16, k17, k18) is added to this value using exclusive OR

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

- Let us rename these 8 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 4 bits (first row of the preceding 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 boxes are defined as follows:

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

- The S-boxes operate as follows.
- The first and fourth input bits are treated as a 2-bit number that specify a row of the S-box, and the second and third input bits specify a column of the Sbox. The entry in that row and column, in base 2, is the 2-bit output. For example, if (p0,0p0,3) = (00) and (p0,1p0,2) = (10), then the output is from row 0, column 2 of S0, which is 3, or (11) in binary. Similarly, (p1,0p1,3) and (p1,1p1,2) 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			
2	4	3	1

- The output of P4 is the output of the function F. The Switch Function The function fK 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 fK operates on a different 4 bits. In this second instance, the E/P, S0, S1, and P4 functions are the same. The key input is K2.