DATA ENCRYPTION STANDARD

It was adopted in 1977 by the National Bureau of Standards (NBS), now National Institute of Standards and Technology (NIST), as Federal Information Processing Standard 46 (FIPS PUB 46). In 1971, IBM's team under Horst Feistel leadership developed algorithm LUCIFER, operating on 64-bit blocks with 128-bit key. Further, IBM's team headed by Walter Tuchman and Carl Meyer revised LUCIFER to make it more resistant to cryptanalysis, but they reduced key size to 56 bits. In 1973, NBS issued a request for proposals for a national cipher standard. IBM submitted results of its Tuchman-Meyer project. This was by far the best algorithm proposed and was adopted in 1977 as Data Encryption Standard. In 1994, NIST reaffirmed DES for federal use for another 5 years. In 1999, NIST issued a new version of its standard (FIPS PUB 46-3) that indicated that DES should only be used for legacy systems and that triple DES be used.

DES ENCRYPTION

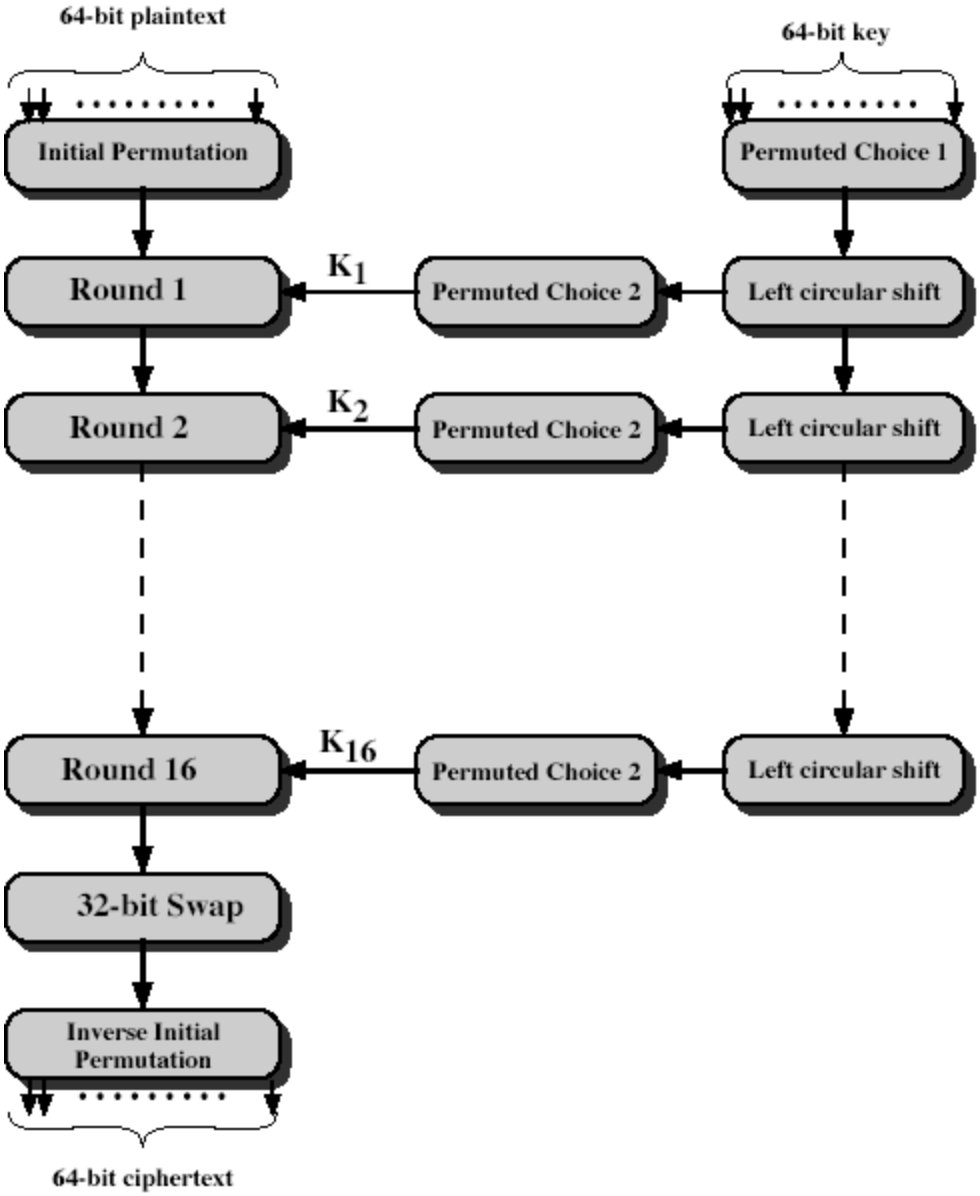

Figure 3.7 General Depiction of DES Encryption Algorithm

32-bit swap swaps left and 32-bit halves obtained after Round 16, we get pre-output. Finally, pre-output passes through a permutation IP^{-1}, that is an inverse to initial permutation IP, to produce the 64-bit cipher-text. The right-hand portion of Fig. 3.7 shows the way in which 56-bit is used. For each of 16 rounds a sub-key Ki is produced by the combination of a left circular shift and a permutation. The permutation function is the same for each round.

INITIAL PERMUTATION AND ITS INVERSE

It affects on 64-bit input

IP
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

IP^{-1}
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

DETAILS OF SINGLE ROUND

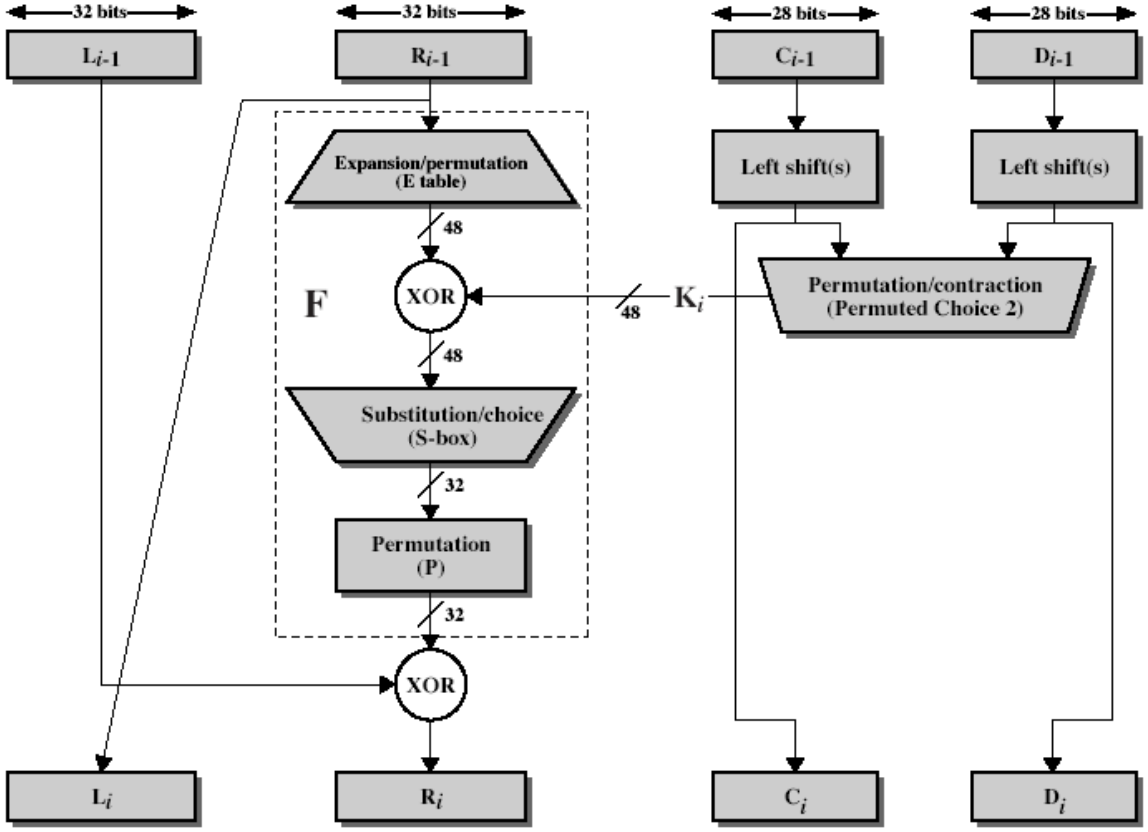

Figure 3.8 Single Round of DES Algorithm

The left and right halves of each 64-bit intermediate value are treated as separate 32-bit quantities, labeled L and R. As in the classic Feistel cipher, the overall process at each round is summarized as follows:

$L_i = R_{i-1}$

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

The round key Ki is 48 bits. The R input is 32 bits. This R input is first expanded to 48 bits by Expansion/Permutation (E table):

Expansion/Permutation (E table)		
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

DETAILS OF SINGLE ROUND (CONT 1)

The resulting 48 bits are XORed with Ki. This 48 bit result passes through a substitution function that produces 32-bit output, which is permuted by Permutation function (P):

Permutation function(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

The role of S-boxes is illustrated in Fig. 3.9:

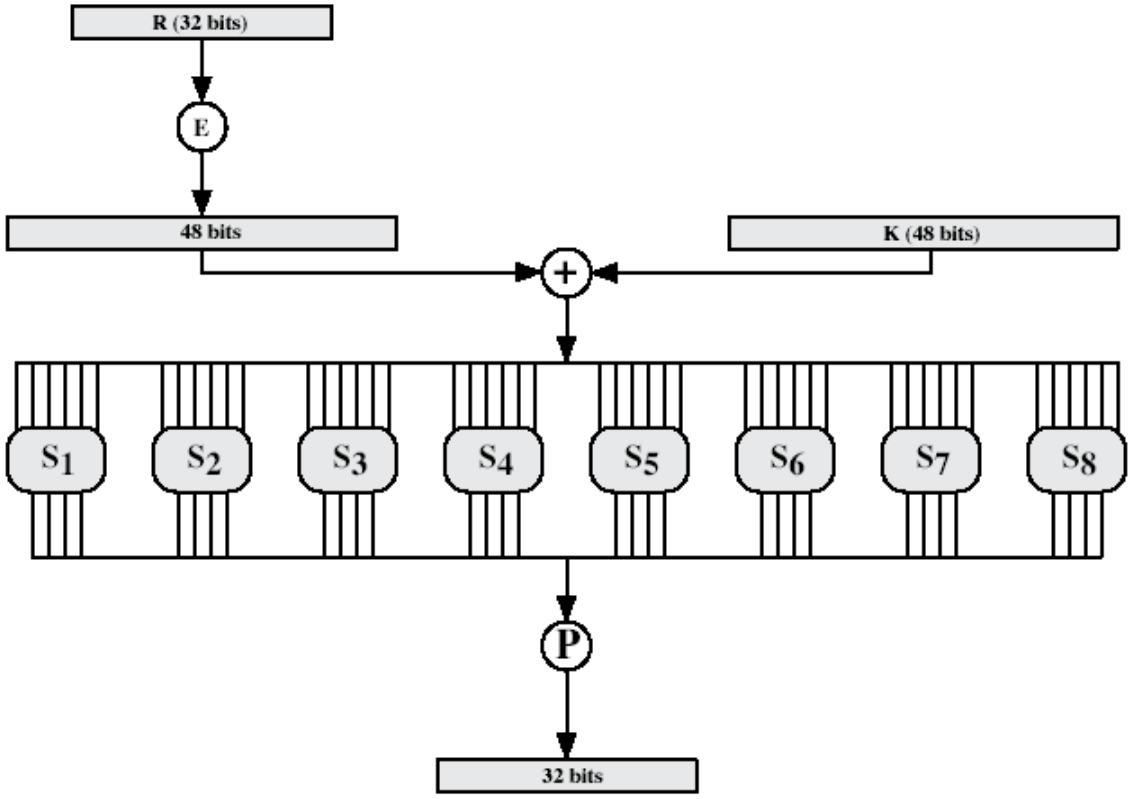

Figure 3.9 Calculation of F(R, K)

The substitution consists of a set of 8 S-boxes, each of which accepts 6 bits input and produces 4 bits as output.

DETAILS OF SINGLE ROUND (CONT 2)

These transformations are:

Table 3.3 Definition of DES S-Boxes

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

Each row of an S-box defines a general reversible substitution: middle 4 bits of each group of 6-bit input are substituted by S-box output, 1st and last 6th bits define what particular substitution out of four to use.

KEY GENERATION

Input key has 64 bits. But each 8th bit is not used: bits 8,16,24,32,40,48,56,64 are not further used. The 56-bit key is first subjected to permutation Permuted Choice 1:

Permuted Choice 1 (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

The resulting 56-bit key is then treated as two 28-bit quantities, labeled C0 and D0. At each round, C_{i-1} and D_{i-1} are separately subjected to a circular left shift, or rotation, of 1 or 2 bits as governed by the following:

Schedule of Left Shifts															
Round number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bits rotated	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

These shifted values serve as input to the next round. They also serve as input to Permuted Choice 2, which produces a 48-bit output that serves as input to the function $F(R_{i-1}, K_i)$.

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

DES DECRYPTION

As with any Feistel cipher, decryption uses the same algorithm as encryption, except that the application of sub-keys is reversed.

THE AVALANCHE EFFECT IN DES

1 bit change in the plaintext leads to 34 bit difference in the cipher-text. 1 bit change in the key leads to 35 bit difference in the cipher-text.

THE STRENGTH OF DES

DES proved insecure in July 1998, when the Electronic Frontier Foundation (EFF) announced that it had broken a DES encryption using a special-purpose "DES cracker" machine that was built for less than $250 000. The attack took less than 3 days.

Design criteria for S-boxes were not made public, so there was a concern that cryptanalysis is possible for an opponent who knows the weaknesses in S-boxes. Up to now, there are no published results about such weaknesses in S-boxes.

DES also appears to be resistant to timing attack but suggest some avenues to explore. Timing attack tries to understand essence of algorithm by analysis of time of its work on different inputs. One of such approaches yields a Hamming weight (number of bits equal to 1) of the secret key.

DIFFERENTIAL AND LINEAR CRYPTANALYSIS

Differential cryptanalysis attack is first published attack that is capable of breaking DES in less than 2^{55} complexity. The scheme can successfully crypt-analyze DES with an effort of the order of 2^{47}, requiring 2^{47} chosen plaintexts. Idea is to follow differences in 2 plaintexts in the rounds of DES transformations, and to estimate probability of the output difference depending on the used key. The first open publication on the differential cryptanalysis was in 1990.

Linear cryptanalysis was described in 1993. Idea is to find linear equation (with XOR operations) between bits of plaintext, cipher-text and key that holds with probability greater than 0.5.

BLOCK CIPHER DESIGN PRINCIPLES

1. No output of any S-box should be too close a linear function of the input bits.
2. Each row of an S-box should include all 16 possible output combinations
3. If two inputs to an S-box differ in exactly 1 bit, the outputs must differ in at least 2 bits.
4. If 2 inputs to an S-box differ in the 2 middle bits exactly, the outputs must differ in at least 2 bits.
5. If 2 inputs to an S-box differ in their first 2 bits and are identical in their last 2 bits, the 2 outputs must not be the same
6. For any nonzero 6-bit difference between inputs, no more than 8 of the 32 pairs of inputs exhibiting that difference may result in the same output difference

These criteria are intended to increase confusion properties.

BLOCK CIPHER DESIGN PRINCIPLES (CONT 1)

The criteria for the permutation P are as follows:

1. The 4 output bits from each S-box at round I are distributed so that 2 of them affect (provide input for) "middle bits" of round (i+1) and the other 2 affect end bits. The 2 middle bits of input to an S-box are not shared by adjacent S-boxes. The end bits are the 2 left-hand bits and the 2 right-hand bits which are shared with adjacent S-boxes (for example, bits 1,2,3,4 – outputs of S1, affect on bits 9, 17 (end-bits) and 23,31 (middle-bits) respectively
2. The 4 output bits from each S-box affect 6 different S-boxes on the next round, and no 2 affect the same S-box (for example, bits 1,2,3,4 - outputs of S1, affect (S2, S3) – bit 1, (S4,S5) – bit 2, S6 – bit 3, S8 – bit 4
3. For 2 S-boxes j, k, if an output bit from Sj affects a middle bit of Sk on the next round, then an output bit from Sk cannot affect a middle bit of Sj. This implies that for j=k, an output bit from Sj must not affect a middle bit of Sj. For example, output bit 3 of S1 affects middle bit of S6. Then we are to have that output bits of S6 are not to affect middle bits of S1. Output bits of S6 are 21,22,23,24. Bit 21 affects bit 4 – end bit of S1, S2; bit 22 affects bit 29 – end bit of S7, S8; bit 23 affects bit 11 – middle bit of S3; bit 24 affects bit 19 – middle bit of S5. So, middle bits of S1 are not affected by output bits of S6.

These criteria are intended to increase diffusion properties.

Number of rounds – the greater this number, the more difficult is cryptanalysis. If DES had 15 or less rounds, differential cryptanalysis would require less effort than brute-force attack

Function F should be nonlinear, provide avalanche effect. Also, bit independence criterion is used: output bits j and k should change independently when any single input bit i is inverted, for all i,j,k.

Size of S-boxes: larger size – more resistant to differential and linear cryptanalysis. S-boxes may be made randomly or according to mathematical rules automatically. Contents of S-boxes may depend on the key.

Key schedule algorithm – no general principles for this have yet been promulgated. Key schedule (production of sub-keys) should guarantee key/cipher-text avalanche criterion and bit independence criterion.

BLOCK CIPHER MODES OF OPERATION

Four DES modes of operations have been defined (FIPS 81, http://www.itl.nist.gov/fipspubs/fip81.htm):

Mode	Description	Typical application
Electronic Codebook (ECB)	Each block of 64 plaintext bits is encoded independently using the same key	Secure transmission of single values (e.g., an encryption key)
Cipher Block Chaining (CBC)	The input to the encryption algorithm is the XOR of the next 64 bits of the plaintext and the preceding 64 bits of the cipher-text	General-purpose block-oriented transmission Authentication
Cipher Feedback (CFB)	Input is processed s bits at a time. Preceding cipher-text is used as input to the encryption algorithm to produce pseudorandom output, which is XORed with plaintext to produce next unit of cipher-text	General-purpose stream-oriented transmission Authentication
Output Feedback (OFB)	Similar to CFB, except that the input to the encryption algorithm is the preceding DES output	Stream-oriented transmission over noisy channel (e.g., satellite communication)
Counter (CTR)	Each block of plaintext is XORed with an encrypted counter. The counter is incremented for each subsequent block	General-purpose block-oriented transmission Useful for high-speed requirements

ELECTRONIC CODEBOOK MODE

It may be considered that each 64-bit plaintext is mapped to respective 64-bit cipher-text, and each such possible pair represents 1 page of the codebook

For lengthy messages ECB mode may be not secure. If the message has repetitive elements with a period of repetition a multiple of 64 bits, then these elements can be identified by the analyst.

CIPHER BLOCK CHAINING MODE

We need that same plaintext block, if repeated, produces different cipher-text blocks. The simple way is CBC mode:

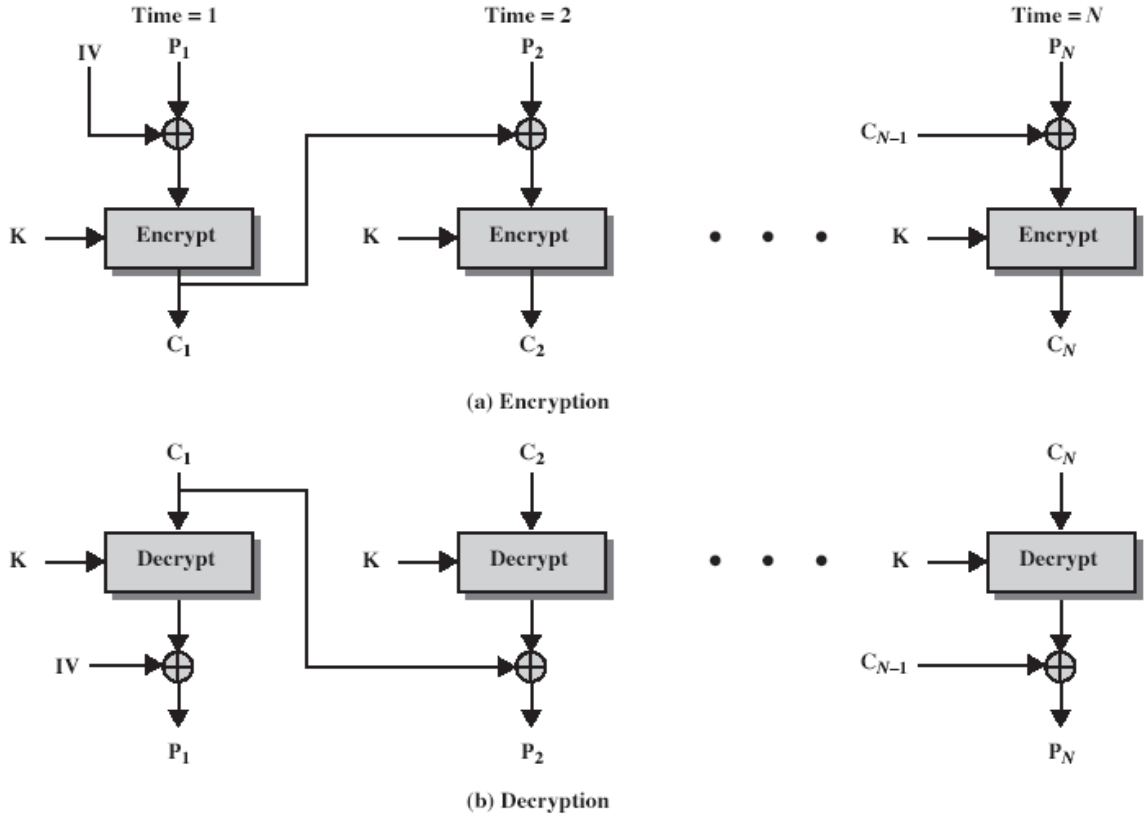

Figure 3.12 Cipher Block Chaining (CBC) Mode

Here IV- Initialization Vector – must be known to both sender and receiver.

CIPHER FEEDBACK MODE

DES is a block cipher, but it may be used as a stream cipher if to use the Cipher Feedback Mode (CFB) or the Output Feedback Mode (OFB). A stream cipher eliminates the need to pad a message to be an integral number of blocks. It also can operate in real time. Thus, if a character stream is being transmitted, each character can be encrypted and transmitted immediately using a character-oriented stream cipher.

CFB scheme follows:

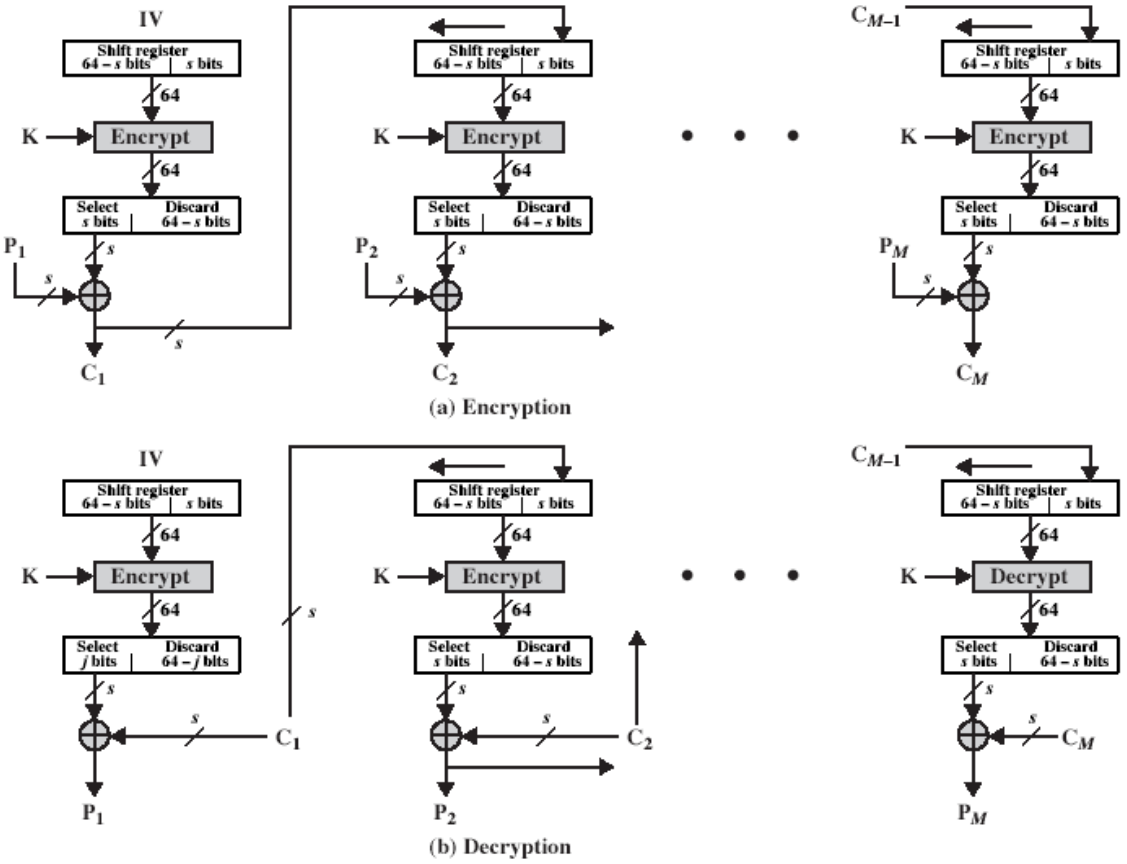

Figure 3.13 *s*-bit Cipher Feedback (CFB) Mode

In Fig. 3.13, it is assumed that the unit of transmission is s bits; usually, s=8. As with CBC, the units of plaintext are chained together, so that the ciphertext of any plaintext unit is a function of all the preceding plaintext. In this case, rather than units of 64 bits, the plaintext is divided into segments of s bits.

Consider encryption. The input to the encryption function is a 64-bit shift register that is initially set to some initialization vector (IV). The leftmost (most significant) s bits of the output of the encryption function are XORed with the first segment of plaintext P1 to produce the first unit of cipher-text C1, which is then transmitted. In addition, the contents of the shift register

CIPHER FEEDBACK MODE (CONT 1)

are shifted left by s bits and C1 is placed in the rightmost (least significant) s bits of the shift register. This process continues until all plaintext units have been encrypted.

For decryption, the same scheme is used except that the received cipher-text unit is XORed with the output of the encryption function to produce the plaintext unit.

OUTPUT FEEDBACK MODE

The Output Feedback Mode (OFB) is similar in structure to that of CFB:

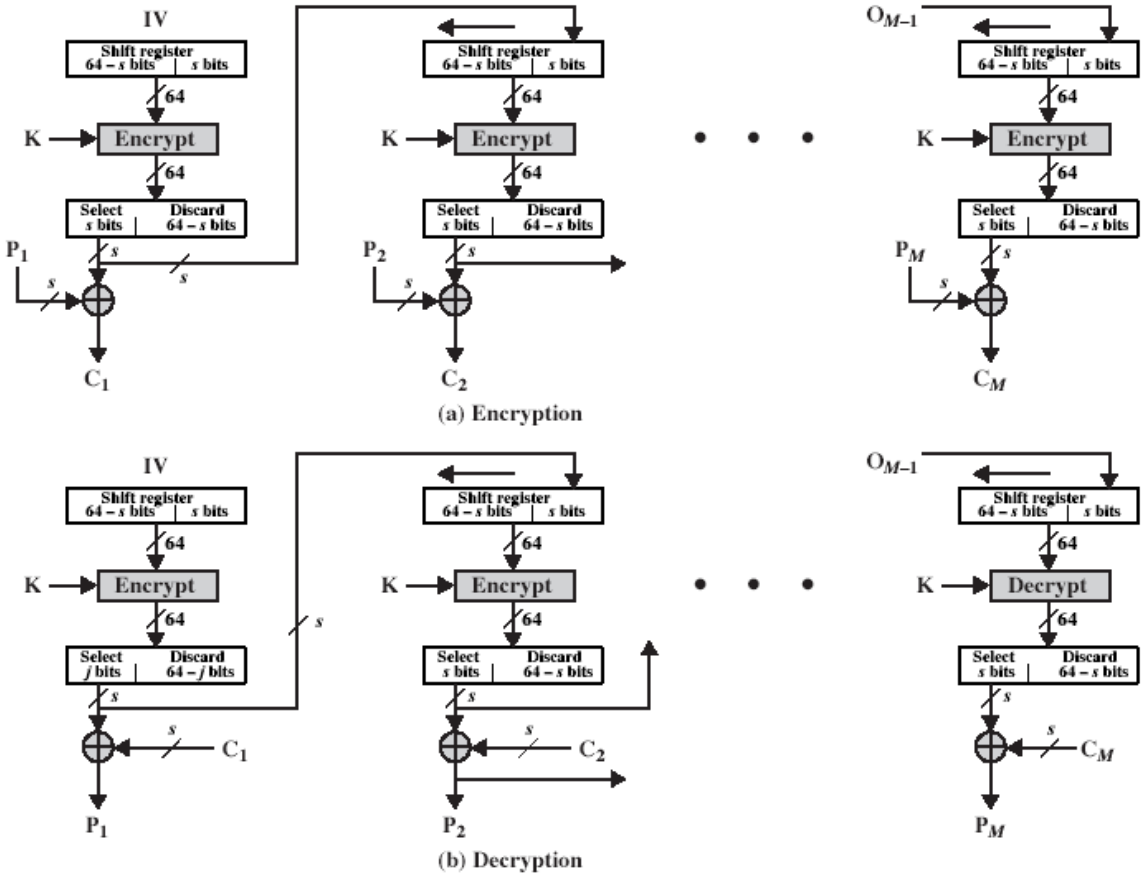

Figure 3.14 *s*-bit Output Feedback (OFB) Mode

As can be seen, it is the output of the encryption function that is fed back to the shift register in OFB, whereas in CFB the cipher-text unit is fed back to the shift register. One advantage of the OFB method is that bit errors in transmission do not propagate.

COUNTER MODE

A counter, equal to the plaintext block size is used. The only requirement stated in SP 800-38 A(http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf, NIST Special Publication 800 -38 A, 2001 Edition, Morris Dworkin, Recommendations for Block Cipher Modes of Operation) is that the counter value must be different for each plaintext block that is encrypted. This mode is with applications to ATM (asynchronous transfer mode) and IPsec (IP security) nowadays, but it was proposed in 1979.
Counter Mode works as follows:

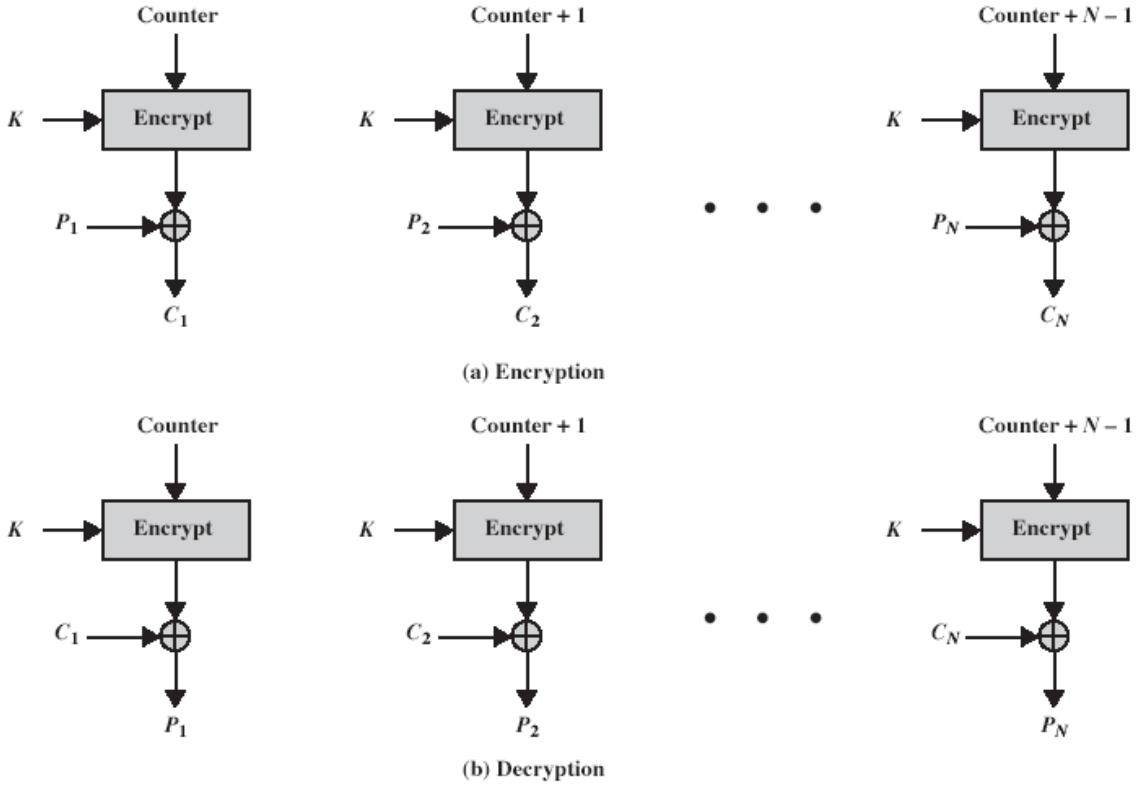

Figure 3.15 Counter (CTR) Mode

Addition is made modulo 2^b, where b is a block size. CTR mode is effective because blocks may be processed in parallel; encryption of keys may be made in advance, and only XOR will be made on-line; only necessary blocks may be decrypted; provides not less security than chaining modes but significantly simpler.