

Chapter Three

Data Encryption Standard

Dr .saad abdual Azize

11/24/2012-2017



Chapter Three

FEISTEL CIPHER STRUCTURE:

Virtually all conventional block encryption algorithms including DES, have a structure first described by HOREST FEISTEL of IBM in 1973 and shown in the following figure:

The input to the encryption algorithm are a plain text block of length $2w$ bits and a key k . The plain text block is divided into two halves, L_0 and R_0 . The two halves of the data pass through n rounds of processing and then combine to produce the cipher text block. Each round i has as inputs L_{i-1} and R_{i-1} , derived from the previous round, as well as a subkey k_i derived from the overall k . In general the subkeys k_i are different from k and from each other and are generated from the key by a subkey generation algorithms.

The exact realization of a Feistel network depends on the choice of the following parameters and design features:

- Block Size: Larger block sizes mean greater security (all other things being equal) but reduced encryption/decryption speed. A block size of 64 bits is a reasonable tradeoff and is nearly universal in block cipher design.
- Key Size: Larger key size means greater security but may decrease encryption/decryption speed. The most common key length in modern algorithms is 128 bits.
- Number of Rounds: The essence of the Feistel cipher is that a single round offers inadequate security but that multiple rounds offer increasing security. A typical size is 16 rounds.
- Subkeys generation algorithm: Greater complexity in this algorithm should lead to greater difficulty of cryptanalysis.
- Round Function: Again, greater complexity generally means greater resistance to cryptanalysis.

There are two other considerations in the design of a Feistel Cipher:

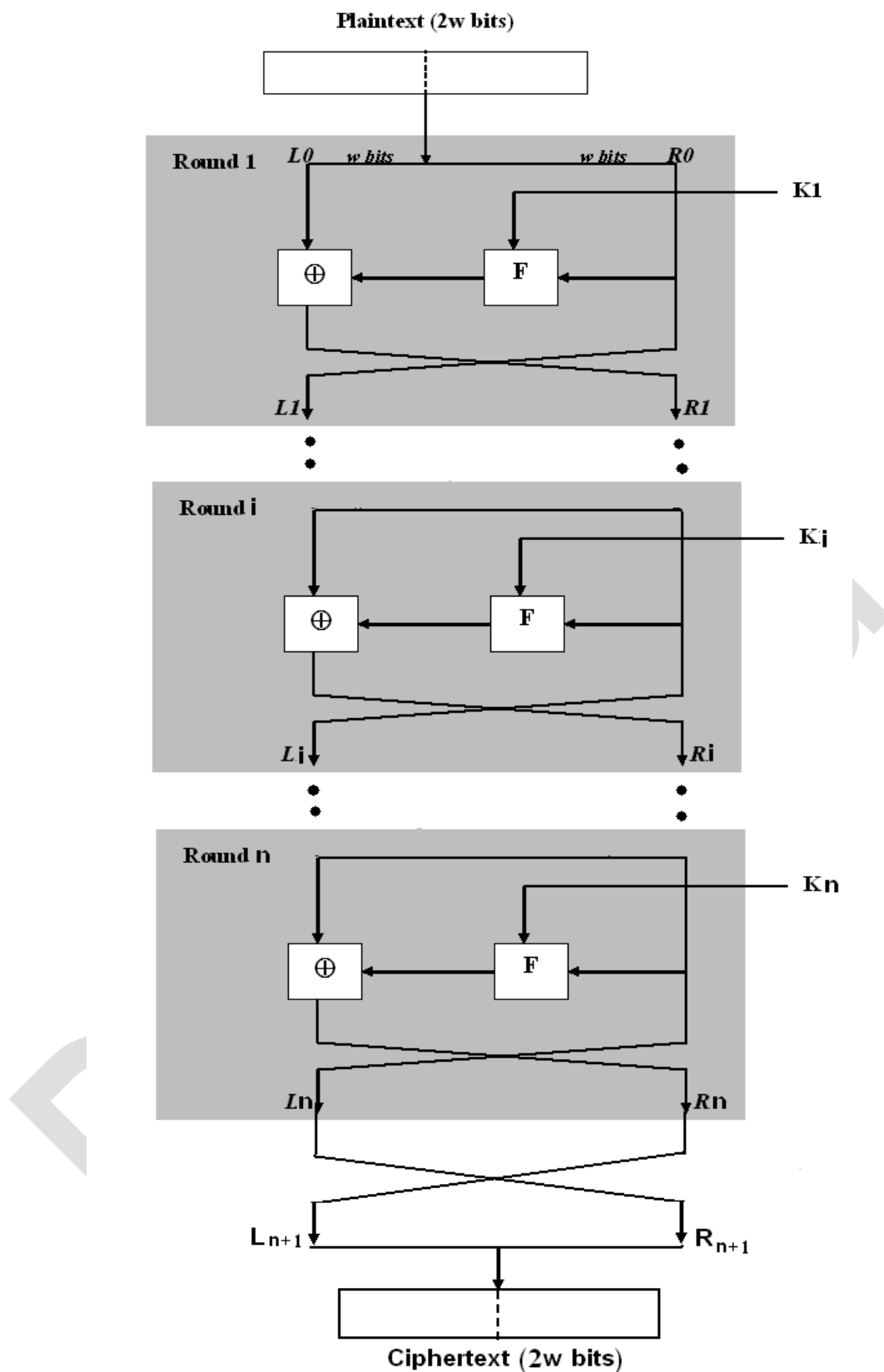
- Fast software encryption/decryption: In many cases, encryption is embedded in applications or utility functions in such away as to preclude a hardware implementation. Accordingly, the speed of execution of the algorithm becomes a concern.



- Ease of Analysis: Although we would like to make our algorithm as difficult as possible to cryptanalysis, there is a great benefit in

Dr. Saad



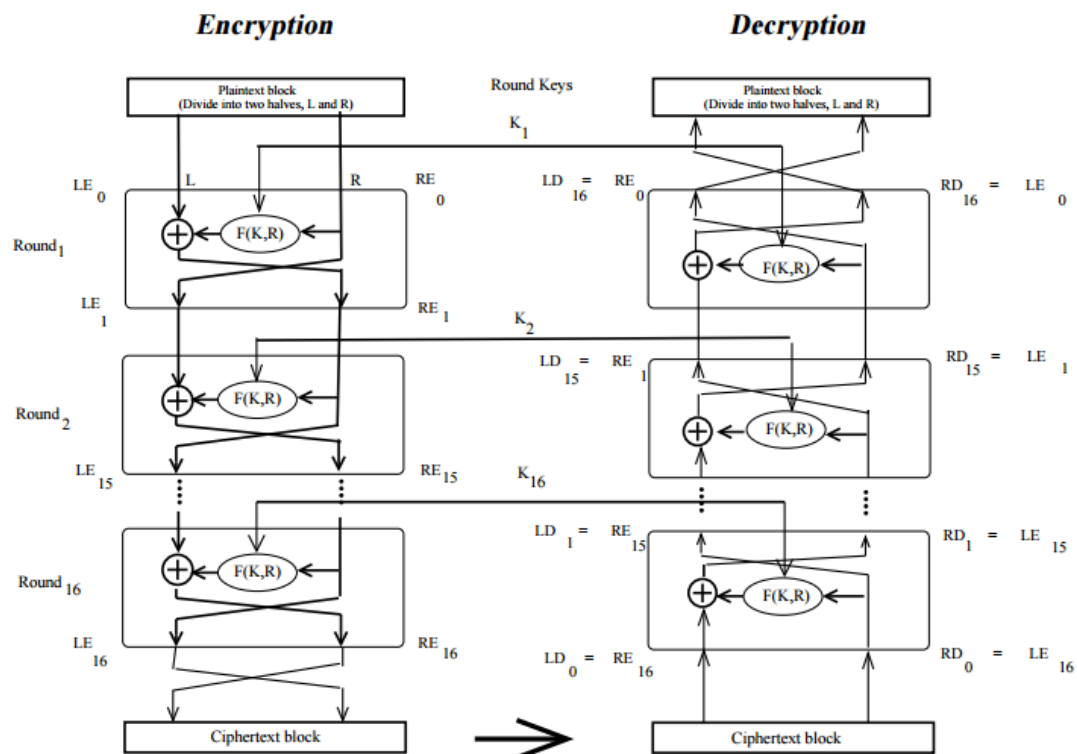


Classical Feistel Network

making the algorithm easy to analyze. That is, if the algorithm can be concisely and clearly explained, it is easier to analyze that algorithm for cryptanalytic vulnerabilities and therefore develop a higher level of assurance as to its strength. DES, for example, does



not have an easily analyzed functionality.



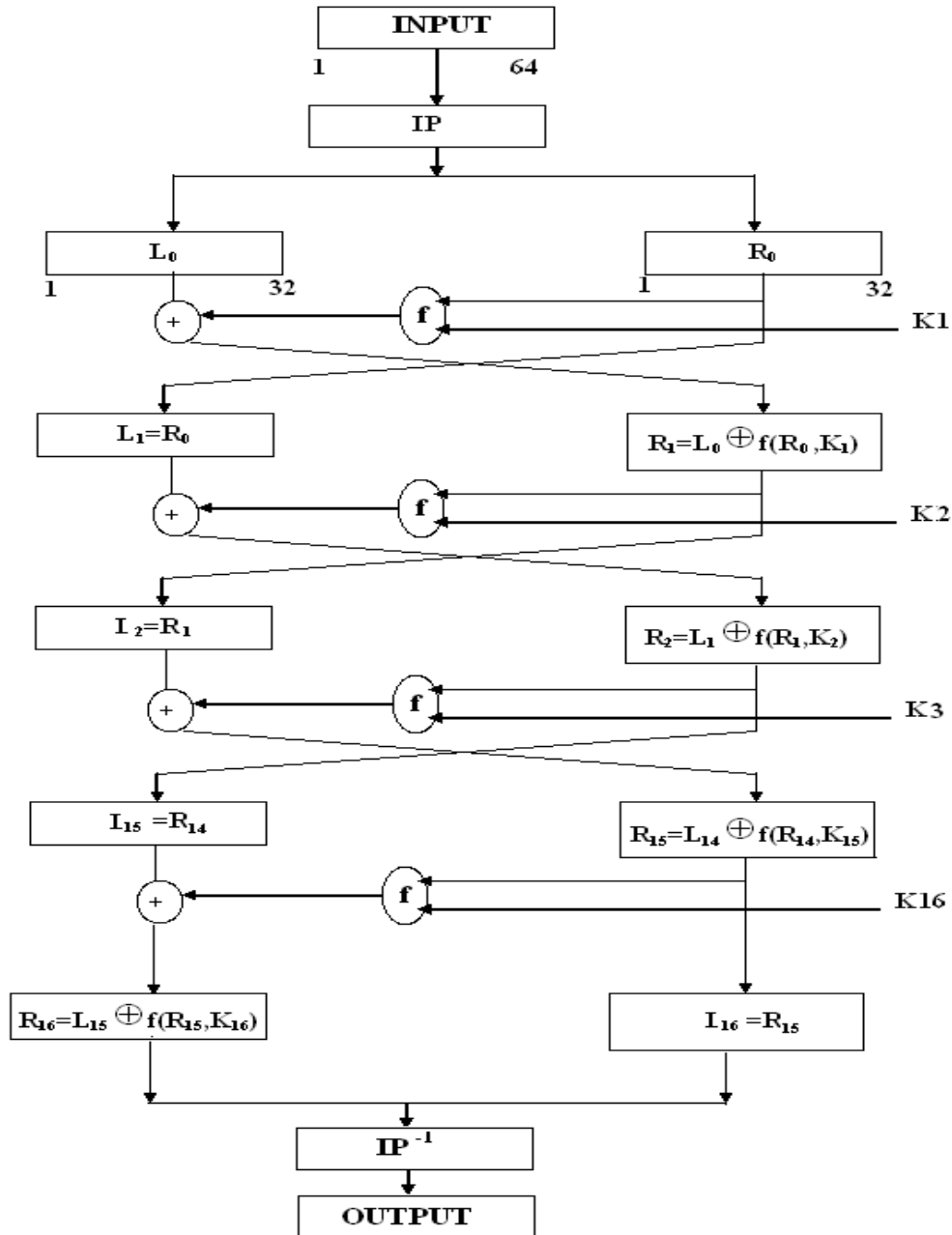
3- DATA ENCRYPTION STANDARD (DES):

The most widely used encryption scheme is defined in the data encryption standard (DES) adopted in 1977 by the National Bureau of standards.

Description of the Algorithm:

The plain text is 64 bits in length and the key is 56 bits in length; longer plain text amounts are processed in 64-bit blocks.





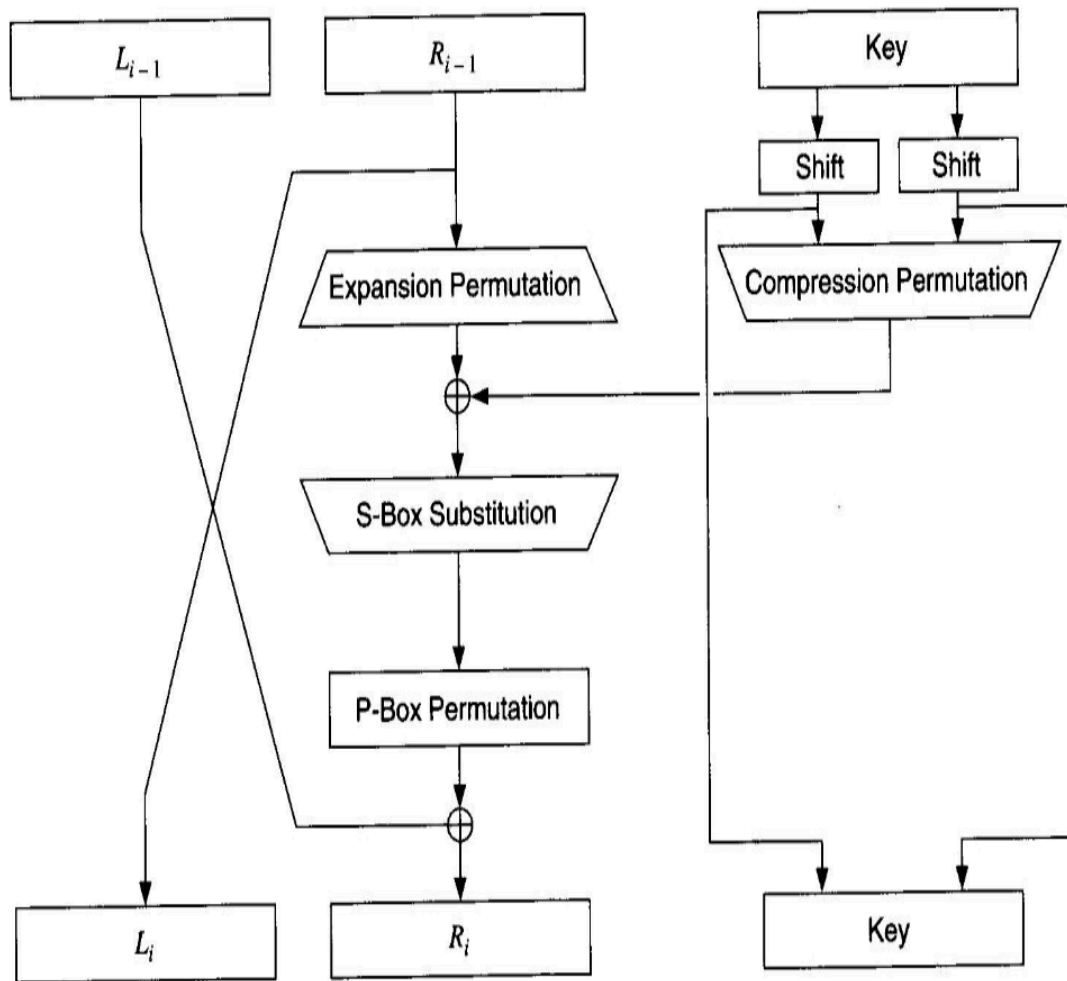


Figure One round of DES in detail

Key



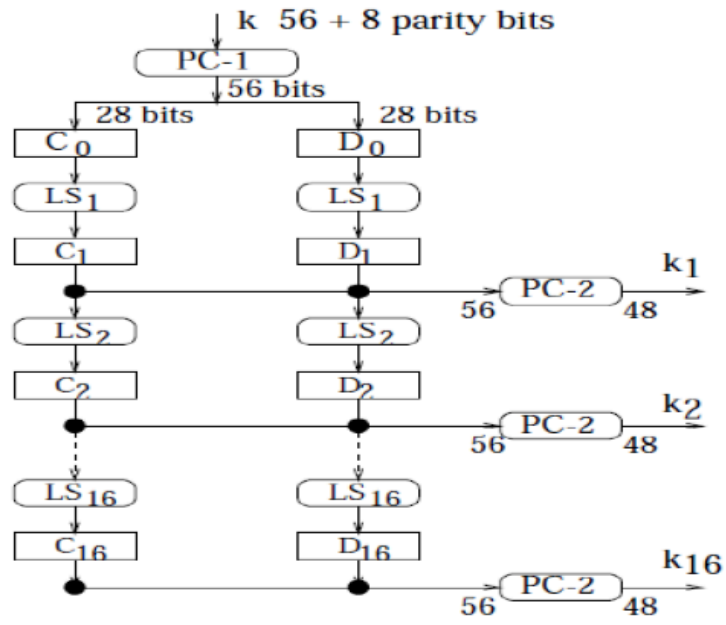


Figure 3. 16th key generation



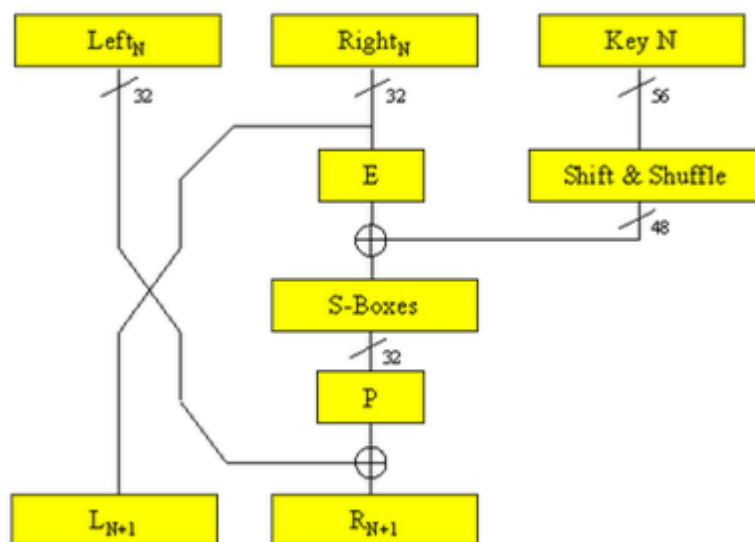
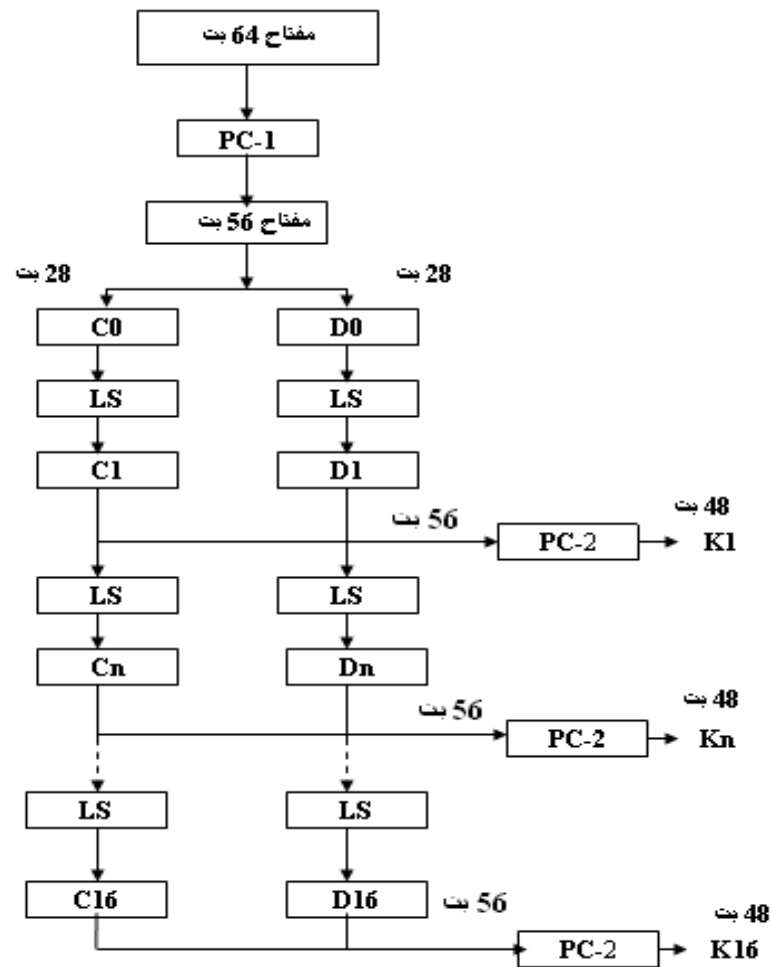


Fig dingle round DES Algorithm

Properties:



1. Each Block consists of 64 bits.
2. Key length is 56 bits (actually 64 bits, 8 bits parity check).
3. The overall processing at each iteration can be summarized in the following formulas:

$$L_i = R_{i-1}$$

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

Where $\oplus$ denotes the bitwise XOR function.

4. DES has 16 iterations.
5. The complex function involves both permutation and substitution operations. The substitution operation, represented as tables called "S-Boxes", simply maps each combination of 48 inputs bits into a particular 32-bit pattern.

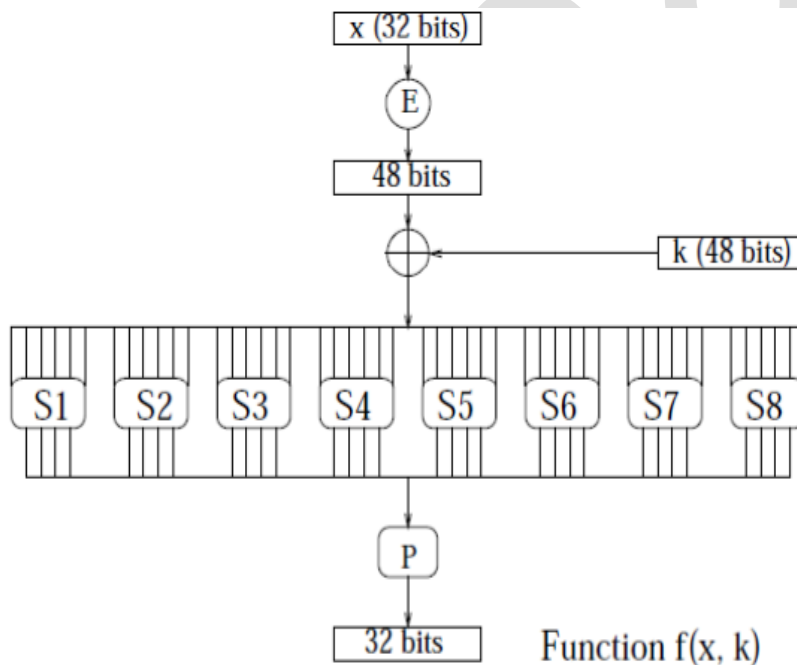
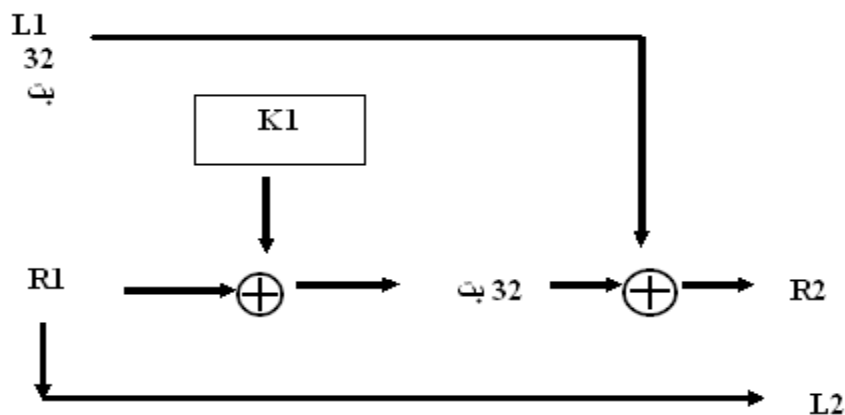


Figure 4. f-function



		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S1	0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S2	0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
	3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
S3	0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
	1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
	2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
	3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
S4	0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
	1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
	2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
	3	3	15	0	6	10	1	13	8	7	4	5	11	12	7	2	14
S5	0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
	1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
	2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
	3	11	8	12	7	1	14	2	13	7	15	0	9	10	4	5	3
S6	0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
	1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
	2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
	3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
S7	0	4	11	2	14	15	12	8	13	3	12	9	7	5	10	6	1
	1	13	0	11	7	4	0	1	10	14	3	5	12	2	5	8	6
	2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
	3	6	1	13	8	1	4	10	7	9	5	0	15	14	2	3	12
S8	0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	1
	1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
	2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
	3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11





S-boxes:

The s-boxes are substitution based on table of 4 rows and 16

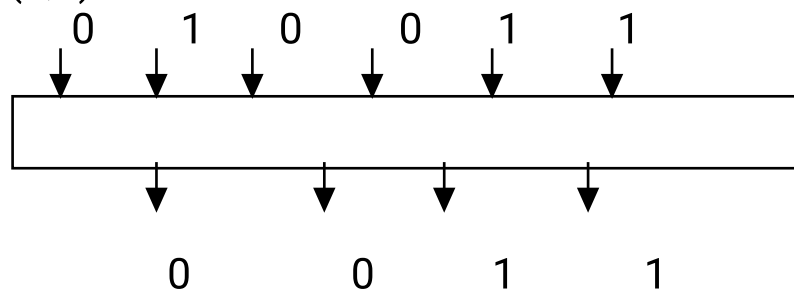
columns. Suppose the block B_j is the six bits $(b_1, b_2, b_3, b_4, b_5, b_6)$. Bits b_1 and b_6 take together, from a two-bit binary number b_1b_6 , having a decimal value from 0 to 3, call this value r . Bits b_2, b_3, b_4 and b_5 taken together from a four-bit binary number $b_2b_3b_4b_5$ having a decimal value from 0 to 15 call the value c . The substitution from the s-boxes transform each 6-bit block B_j into the 4-bit result show in row r , column c of section S_j of table.

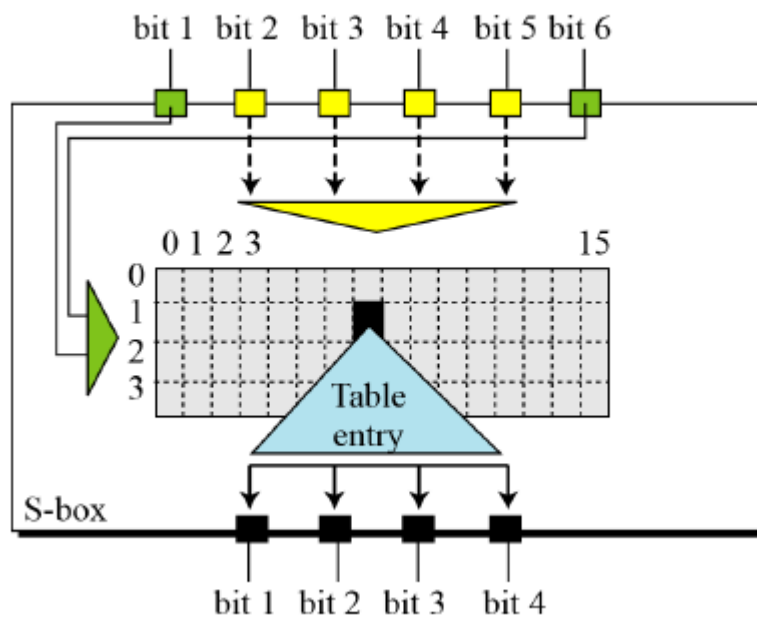
Ex

Assume S_7 in a binary is 010011

Then $r = 01 = 1$ $c = 1001 = 9$

The transformation of block B_7 is found in row 1 and column = $S_7(1,9) = 3 = 0011$





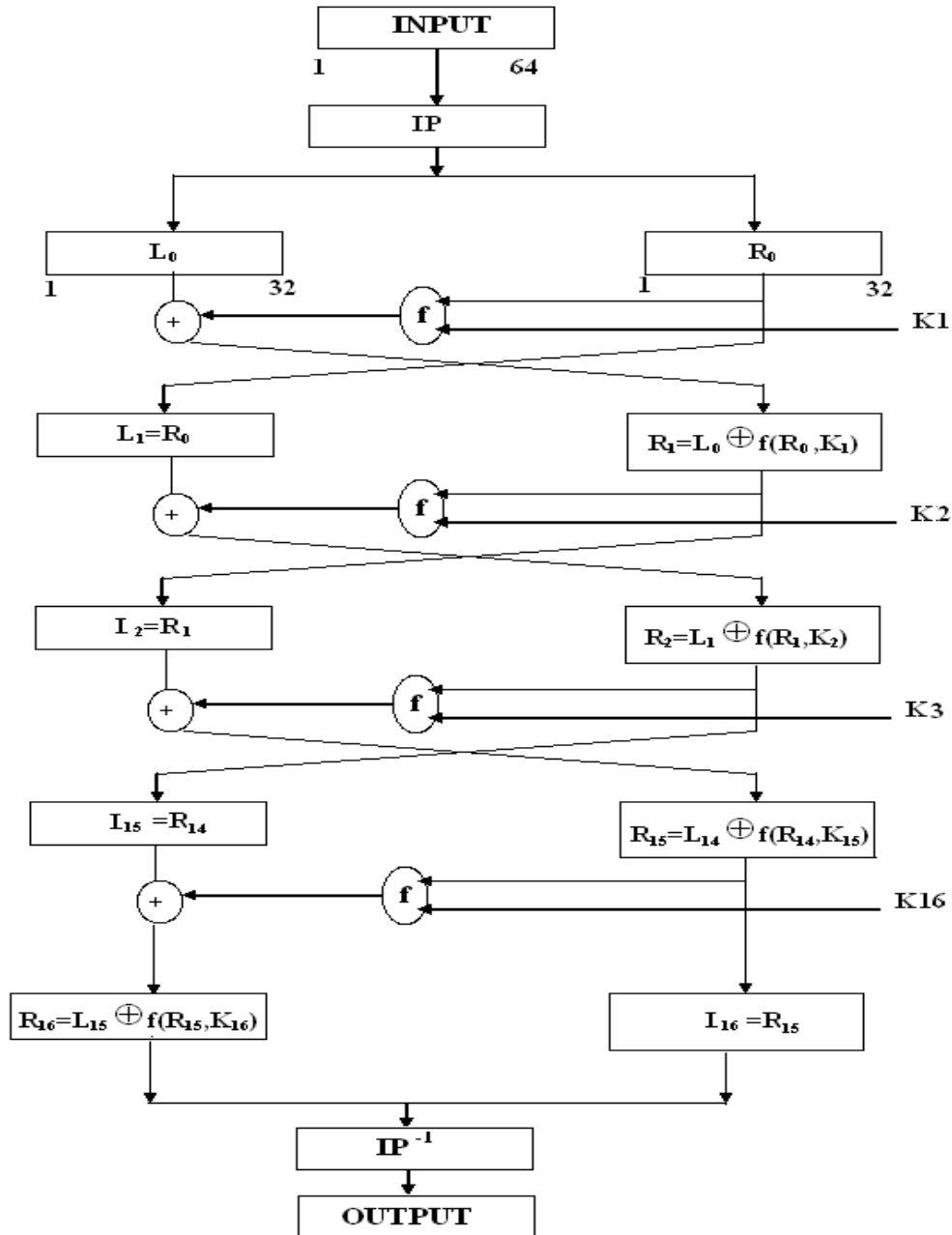
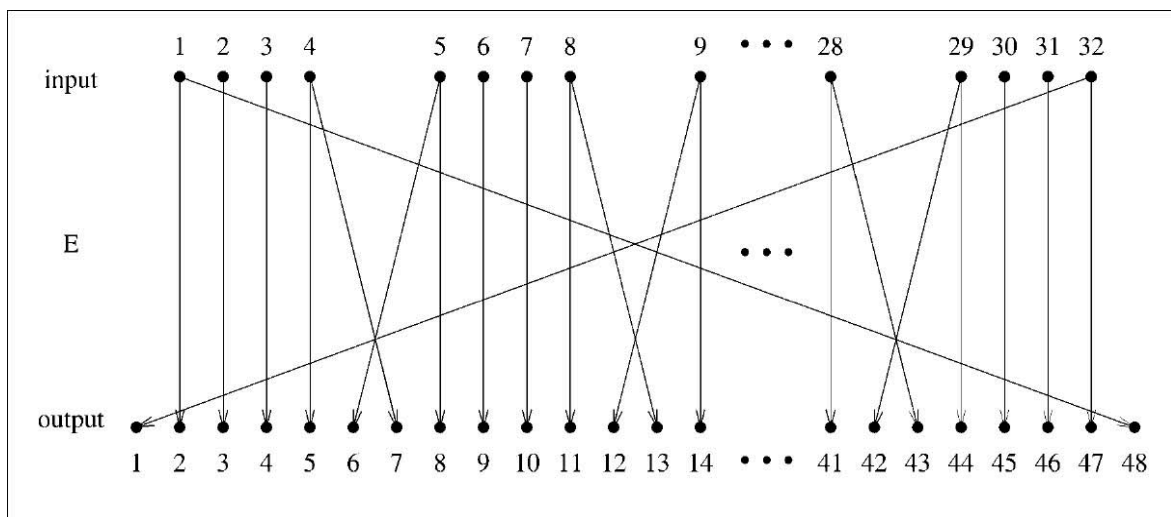


Table 1: Initial IP and inverse initial IP^{-1} permutation tables.

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



E-Expansion Permutation (8,6)

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

PC-1



57	49	41	33	25	17	9	C0
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	D0
7	62	54	46	38	30	22	
14	6	61	53	45	37	29	
21	13	5	28	20	12	4	

DES ALGORITHM – GENERATION OF KEY

The flowchart illustrates the iterative process of generating 16 subkeys from a 64-bit master key. The master key is first permuted by PC1 into two 28-bit halves. Each half is then shifted left by one position (indicated by red boxes with '<<<'). The resulting 28-bit halves are then permuted by PC2 to produce a 48-bit subkey. This cycle is repeated 15 times to generate all 16 subkeys.

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

PC-1

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

PC-2

Screencast-O-Matic.com

Left Shift (LS(



WPS Office تعديل من خلال

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

الإ زاحة	رقم الدورة
1	1,2,9,16
2	3,4,5,6,7,8,10
2	15,14,13,12,11

PC-2 Permuted Choice-2 (8,6)

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

Permntation P(4,8)



Ex :broadcast "COMPUTER" on IP⁻¹ table

1 2 3 4 5 6 7 8
C = 67 = 0 1 0 0 0 0 1 1
9 0 1 2 3 4 5 16
O = 79 = 0 1 0 0 1 1 1 1

M = 77 = 1001101
P = 80 = 1010000
U = 85 = 1010101
T = 84 = 1010100
E = 69 = 1000101
R = 82 = 1010010

64	32	16	8	4	2	1
= 1	0	0	0	0	1	1
						67

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

Output

11011100/01010010/10111100/00010100/10100011/00
000000/11111111/00000000

Ex generate the first and second DES sub keys given that the output of the permuted choice1

1 2 3 4 5 6 7 8 9 16 17 24 25 32 33 40
01 111111 10101010 01010101 10001111
00011111



41 48 49 56 57 64
 01000000 00000001 00001011

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

C0= 0000101 0001101 0100000 0110001
 D0= 1001101 1000111 0110011 0110101
 Shift left one position
 C1= 000101 0001101 0100000 01100010
 D1= 001101 1000111 0110011 01101011

 C1=0001010 0011010 1000000 1100010

The Avalanche Effect:

A desirable property of any encryption algorithm is that a small change in either the plain text or the key should produce a significant change in the cipher text. In particular, a change in one bit of the plain text or one bit of the key should produce a change in many bits of the cipher text. If the change were small, this might provide a way to reduce the size of the plain text or key space to be searched.

Double DES:

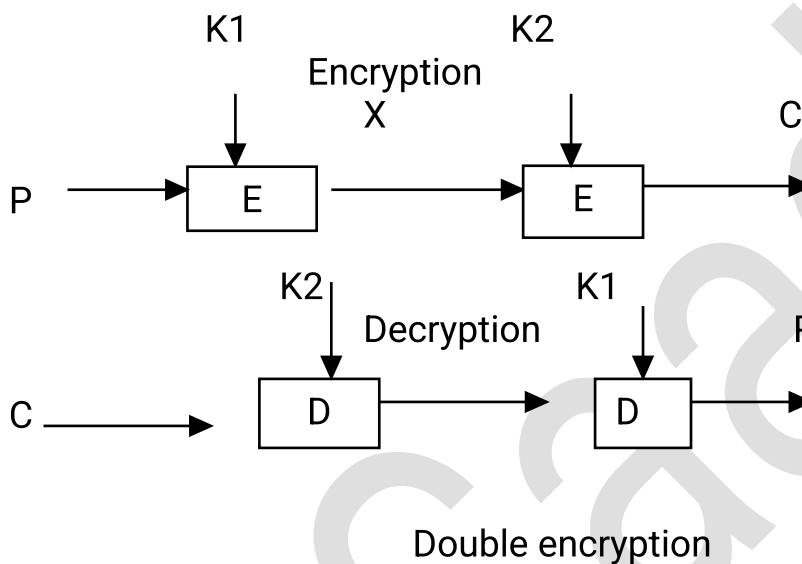
The simplest form of multiple encryption has two encryption stages and two keys. Given a plain text P and two encryption keys k1 and k2, cipher text C is generated as:



$$C = E_{k2} [E_{k1} [P]]$$

Decryption requires that the keys be applied in reverse order:

$$P = E_{k1} [D_{k2} [C]]$$



TRIPLE DEA:

Triple DEA (TDEA) was first proposed by Tuchman. TDEA was incorporated as part of the data encryption standard in 1990. TDEA uses three keys and three executions of the DES algorithm. The function follows an encrypt –decrypt-encrypt (EDE) sequence:

$$C = E_{k3} [D_{k2} [E_{k1} [P]]]$$

Where:

C = Cipher text.

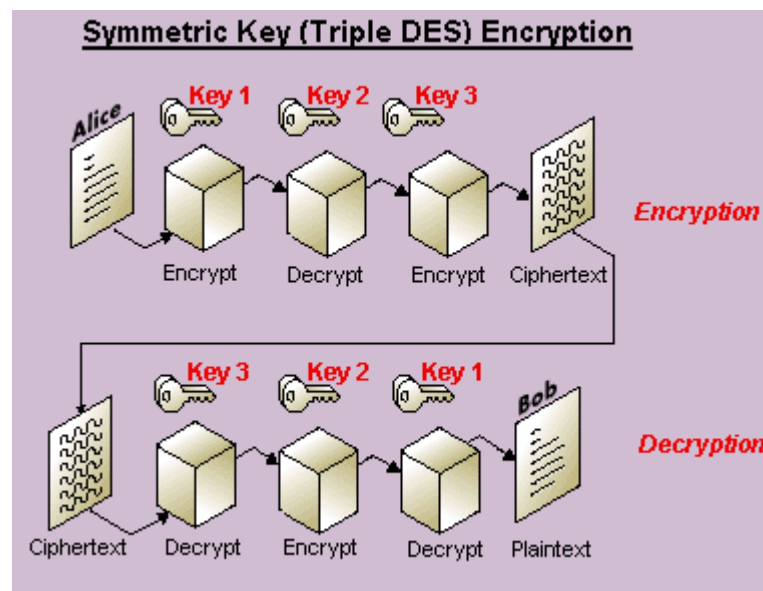
P = Plain text.

$E_k [x]$ = Encryption of X using key K.

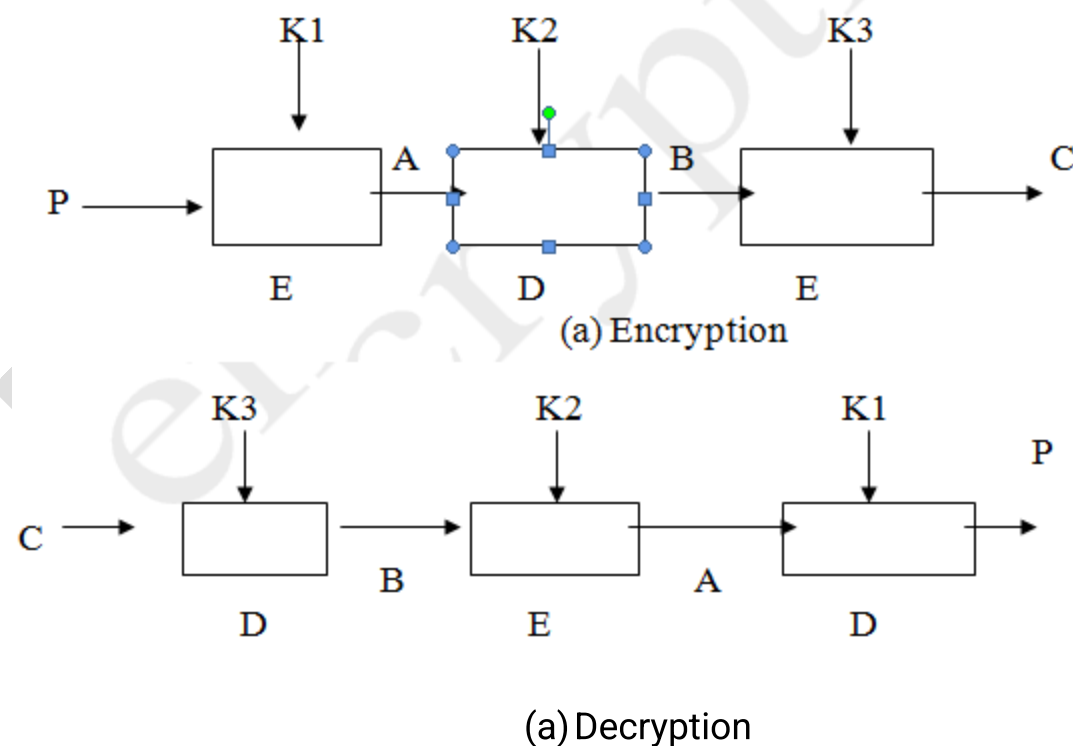
$D_k [y]$ = Decryption of y using key K.

Decryption is simply the same operation with the keys reversed:





$$P = D_{k1} [E_{k2} [D_{k3} [C]]]$$



With three distinct keys, TDEA has an effective key length of 168 bits. It also allow for the use of two keys, with $k_1 = k_3$; this provides for a key length of 112 bits.

TDEA has two attractions:



- 1- With its 168-bit key length, it overcomes the vulnerability to brute-force attack of DEA.
- 2- The underlying encryption algorithm in TDEA is the same as in DEA.

As an alternative, Tuchman proposed a triple encryption method that uses only two keys. The function follows an encrypt-decrypt-encrypt (DES) sequence.

$$C = E_{k1} [D_{k2} [E_{k1} [P]]] = E_{k1} [P]$$

There is no cryptographic significance to the use of decryption for the second stage. Its only advantage is that it allows users of 3DES to decrypt data encrypted by users of the older single DES:

$$C = E_{k1} [D_{k2} [E_{k1} [P]]] = E_{k1} [P]$$

3 DES with two keys is a relatively popular alternative to DES and has been adopted for use in the key management standards ANSI X9.17.

The International Data Encryption Algorithm (IDEA):

Is a symmetric block cipher developed by Swiss Federal Institute of Technology in 1991.

- IDEA uses a 128-bit key.
- IDEA differs from DES both in the round function and in the subkey generation function.
- IDEA does not use S-Boxes.
- IDEA relies on three different mathematical operations: XOR, binary addition of 16-bit integers, and binary multiplication of 16-bit integers.
- The subkey generation algorithm relies solely on the use of circular shifts but uses these in a complex way to generate a total of six subkeys for each of the eight rounds of IDEA.

BLOWFISH:

Blowfish was developed in 1993 by Bruce Schneier. Blowfish was designed to be easy to implement and to have a high



execution speed.

- 1- It is very compact algorithm that can run in less than 5 k of memory.
- 2- Key length is variable and can be as long as 448 bits. In practice, 128-bit keys are used.
- 3- Blowfish uses 16 rounds.
- 4- Uses S-Boxes and the XOR function, also uses binary addition.
- 5- Uses dynamic S-Boxes that are generated as a function of the key.
- 6- A total of 521 executions of the blowfish encryption algorithm are required to produce the subkeys and S-Boxes.
- 7- Fast: Blowfish encrypts data on 32-bit microprocessors at a rate of 18 clock cycles per byte.
- 8- Simple: Blowfish's simple structure is easy to implement and eases the task of determining the strength of the algorithm.

CAST-128:

It is a symmetric encryption algorithms developed in 1997. Its characteristics:

- 1-plaintext $m_1...m_{64}$
- 2-ciphertext $c_1...c_{64}$.
- 3-Key size varies from 40 bits to 128 bits in 8-bits increments.
key $K = k_1...k_{128}$.
- 4-Uses fixed S-Boxes but longer than those used in DES.
- 5-Round function, F , differs from round to round, again adding to cryptanalytic strength.

KEY DISTRIBUTION:

For conventional encryption to work, the two parties to a secure exchange must have the same key, and that key must be protected from access by others.



The strength of any cryptographic system rests with the key distribution technique, a term that refers to the means of delivering a key to two parties that wish to exchange data without allowing others to see the key.

For two parties A and B:

- 1- A key could be selected by A and physically delivered to B.
- 2- A third party could select the key and physically deliver it to A and B.
- 3- If A and B have previously and recently used a key, one party could transmit the new key to the other, encrypted using the old key.
- 4- If A and B each have an encrypted connection to a third party C, C could deliver a key on the encrypted links to A and B.

Options 1 and 2 call for manual delivery of a key. For a link encryption, this is a reasonable requirement. Option 3 is a possibility for either link encryption or end-to-end encryption, but if an attacker ever succeeds in gaining access to one key, then all subsequent keys are revealed. Option 4: the following figure satisfies this option for end-to-end encryption.

For this Schemes, two kinds of keys are identified:

- 1- Session Key: When two end systems (hosts, terminals, etc) wish to communicate, they establish a logical connection (e.g, virtual circuit). For the duration of that logical connection, all user data are encrypted with a one time session key. At the conclusion of the session, or connection, the session key is destroyed.
- 2- Permanent Key: A permanent key is a key used between entities for the purpose of distributing session keys.
The configuration consists of the following elements:
 - a- Key Distribution Centre: It determines which systems are allowed to communicate with each other. When permission is granted for two systems to establish a connection , the key distribution centre provides a one-time session key for that connection.
 - b- Front-End Processor: The front end processor performs end-to-end encryption and obtains session keys on behalf of its



host or terminal.

The Use of 56-Bit Keys

With a key length of 56 bits, there are possible keys, which is approximately keys. Thus, on the face of it, a brute-force attack appears impractical.

Assuming that, on average, half the key space has to be searched, a single machine performing one DES encryption per microsecond would take more than a thousand years (see Table 2.2) to break the cipher.

However, the assumption of one encryption per microsecond is overly conservative. As far back as 1977, Diffie and Hellman postulated that the technology existed to build a parallel machine with 1 million encryption devices, each of which could perform one encryption per microsecond [DIFF77]. This would bring the average search time down to about 10 hours. The authors estimated that the cost would be about \$20 million in 1977 dollars.

استخدام مفاتيح 56 بت
مع طول مفتاح من 56 بت، وهناك مفاتيح ممكنة، وهو ما يقرب من مفاتيح. وهكذا، يبدو أن هجوم القوة الغاشمة، على وجهه، غير عملي. على افتراض أنه في المتوسط، نصف مساحة رئيسية لابد من البحث، آلة واحدة أداء واحد ديس التشفير لكل ميكروثانية سوف يستغرق أكثر من ألف سنوات (انظر الجدول 2.2) لكسر الشفرات. ومع ذلك، فإن افتراض تشفير واحد لكل ميكروثانية هو المحافظ بشكل مفرط. منذ عام 1977، يفترض ديفي وهيلمان أن هذه التكنولوجيا موجودة لبناء آلة موازية مع 1 مليون أجهزة التشفير، كل منها يمكن إجراء تشفير واحد لكل ميكروثانية [DIFF77]. وهذا من شأنه أن يجلب متوسط وقت البحث إلى حوالي 10 ساعات. ويقدر المؤلفون أن التكلفة سيكون حوالي 20 مليون دولار في عام 1977 دولار.

Number of Rounds

The cryptographic strength of a Feistel cipher derives from three aspects of the design: the number of rounds, the function F , and the key schedule algorithm. Let us look first at the choice of the number of rounds.

عدد الجولات
قوة التشفير من الشفرات فيستيل مشتق من ثلاثة جوانب من تصميم: عدد من جولات، وظيفة F ، وخوارزمية الجدول الزمني الرئيسي. دعنا ننظر أولاً في اختيار عدد من جولات.

The greater the number of rounds, the more difficult it is to perform cryptanalysis, even for a relatively weak F . In general, the criterion should be that the number of rounds is chosen so that known cryptanalytic efforts require greater effort than a simple brute-force key search attack. This criterion was certainly used in the design of DES. Schneier [SCHN96] observes that for 16-round DES, a differential cryptanalysis attack is slightly less efficient than brute force: The differential cryptanalysis attack requires operations, 10 whereas brute force requires . If DES had 15 or fewer rounds, differential cryptanalysis would require less effort than a brute-force key search.



وكلما زاد عدد الجولات، كلما كان الأمر أكثر صعوبة في إجراء تحليل الشفرة، حتى بالنسبة للضعفاء نسبيا F. عموما، ينبغي أن يكون المعيار أن يتم اختيار عدد من الجولات بحيث تتطلب جهود التشفير المعروفة أكبر جهد من مجرد القوة الغاشمة هجوم البحث الرئيسي. هذا المعيار كان يستخدم بالتأكيد في تصميم ديس. ويشير سنير [SCHN96] أنه لمدة 16 جولة ديس، التفاضلية هجوم كريبتاناليسيس أقل قليلا كفاءة من القوة الغاشمة: التفاضلية تتطلب هجوم كريبتاناليسيس العمليات، 10 في حين تتطلب القوة الغاشمة. إذا كان ديس 15 جولات أو أقل، والتحليل الكمي التفاضلي يتطلب جهدا أقل من البحث الرئيسي القوة الغاشمة.

Dr. Saad

