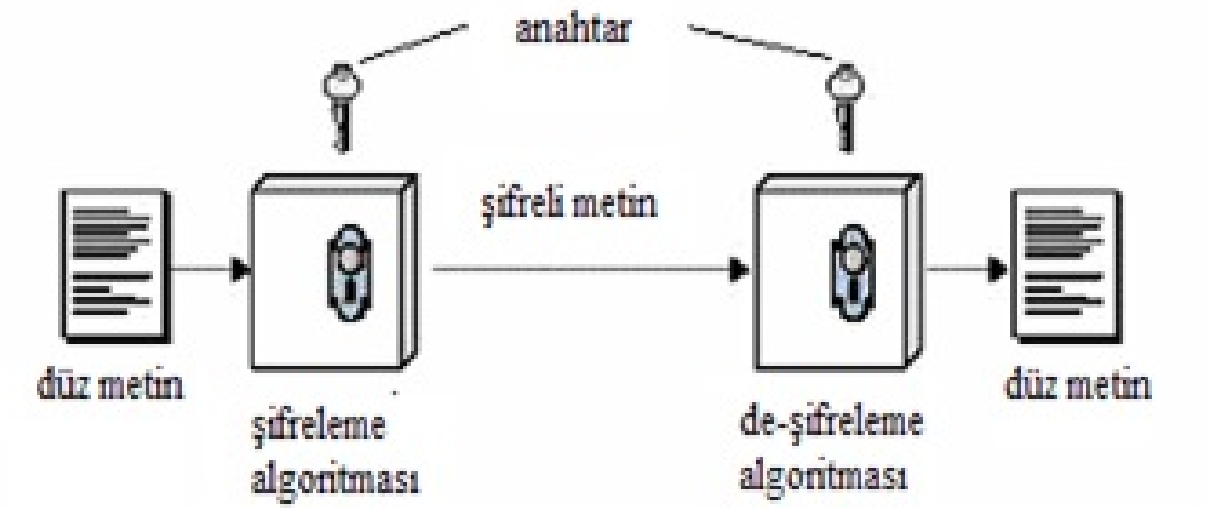
DES ile Veri Şifreleme

Şifreleme gönderilecek bir mesajın gizliliğini başkalarından sakındırmak için kullanılan bir yöntemdir. Şifreleme çeşitlerinden biri olan simetrik şifrelemede amaç veriyi yollayan ile veriyi alan aynı anahtar üzerinde ve aynı şifreleme algoritması ile deşifreleme algoritması üzerinde anlaşıp, mesajı istenmeyen kişilerden korumaktır.

Simetrik şifrelemelerde genel olarak beş bileşen vardır. Bu bileşenler ve şifreleme işleminin nasıl gerçekleştiği şekil de gösterilmiştir.



Simetrik şifrelemede güvenliği sağlayan anahtardır. Çünkü gizli olan tek şey anahtardır. Şifreleme ve deşifreleme algoritmaları herkese açıktır. Farklı anahtarlar sayesinde aynı mesaj ve aynı algoritma ile birbirinden bağımsız şifreli metinler üretilebilir.

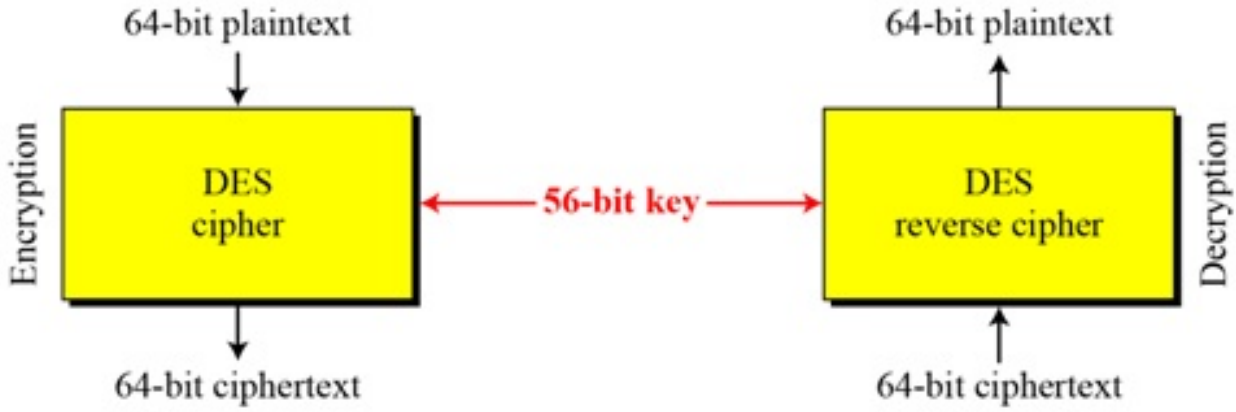
Simetrik şifreleme yöntemleri metin üzerindeki işlemlerine göre iki grup altında sınıflandırılabilir. Bunlardan biri karakter tabanlı yani geleneksel şifreleme sistemleri (Monoalfabetik ve polialfabetik) ve diğeri ise bit tabanlı şifreleme yani modern şifreleme sistemleridir. Bit tabanlı şifreleme sistemlerine örnek olarak DES, IRON ve AES verilebilir.

DES ALGORİTMASI

DES algoritması bir Blok Şifreleme algoritmasıdır. Yani şifrelenecek metin bloklar halinde şifreleme işleminden geçirilir. Ayrıca DES algoritması simetrik şifreleme prensibine dayanmaktadır. Yani DES, veri bloklarını şifrelemek ve deşifrelemek için aynı

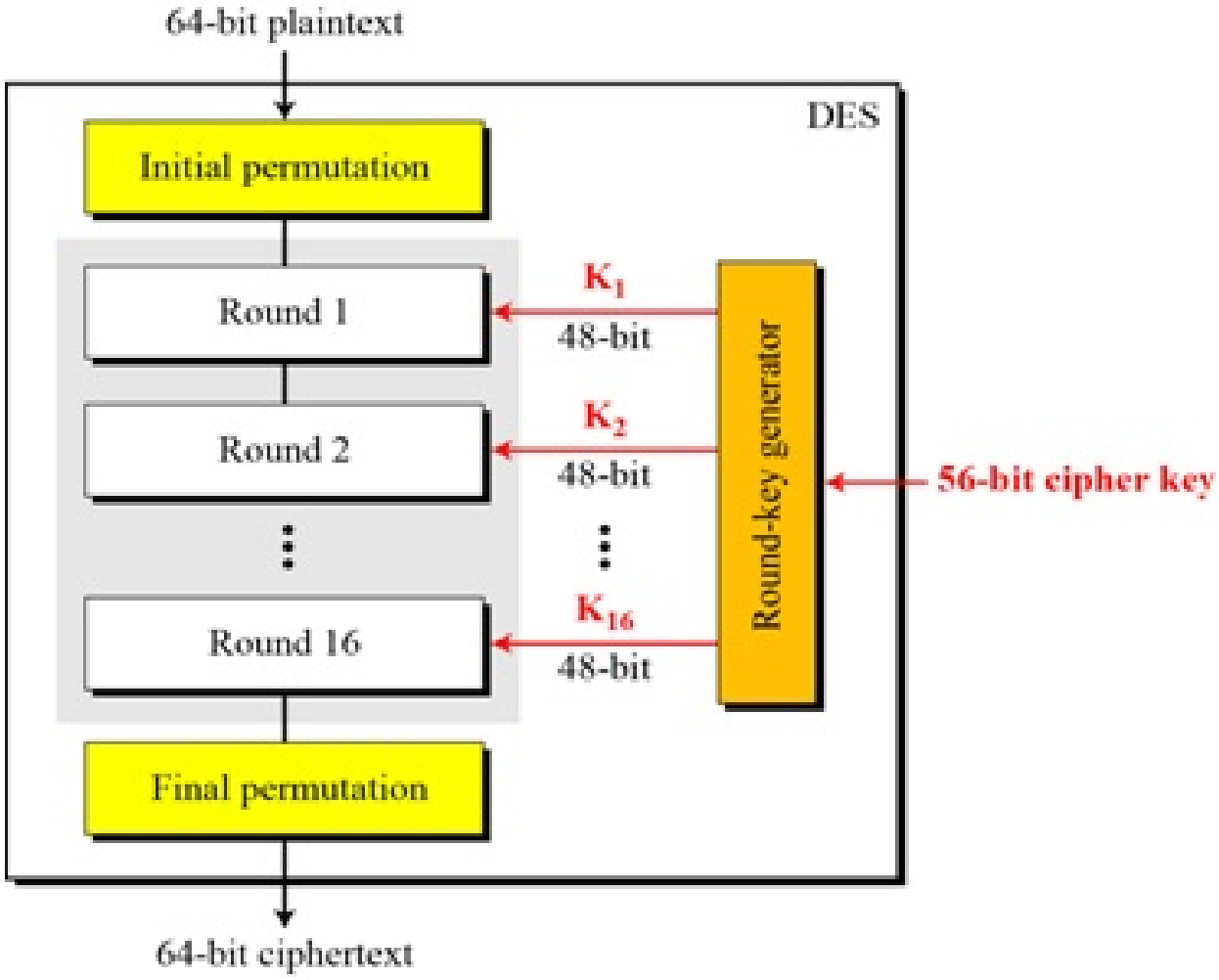
anahtarları kullanmaktadır. DES 64 Bitlik düz metin blokları üzerinde işlem yapmaktadır.

64 bitlik veri blokları, 56 bitlik bir anahtarın kontrolünde şifrelenerek yine 64 bitlik şifrelenmiş metin bloklarına dönüştürülür. Deşifrelenirken de 64 bitlik şifrelenmiş veri blokları, 56 bitlik bir anahtarın kontrolünde deşifrelenerek yine 64 bitlik deşifrelenmiş metinlere (düz metne) dönüştürülür.



DES şifreleme algoritmasının iç yapısına inersek aşağıda verilen şekil karşımıza çıkacaktır. Bu tablodan DES simetrik şifreleme

algoritmasının yapısını daha iyi kavramış oluruz.



Resimde DES algoritmasının genel yapısı verilmiştir. Bu yapıya göre DES algoritması

nasıl çalıştığını yazdığımız kaynak kod ile birlikte açıklayalım:

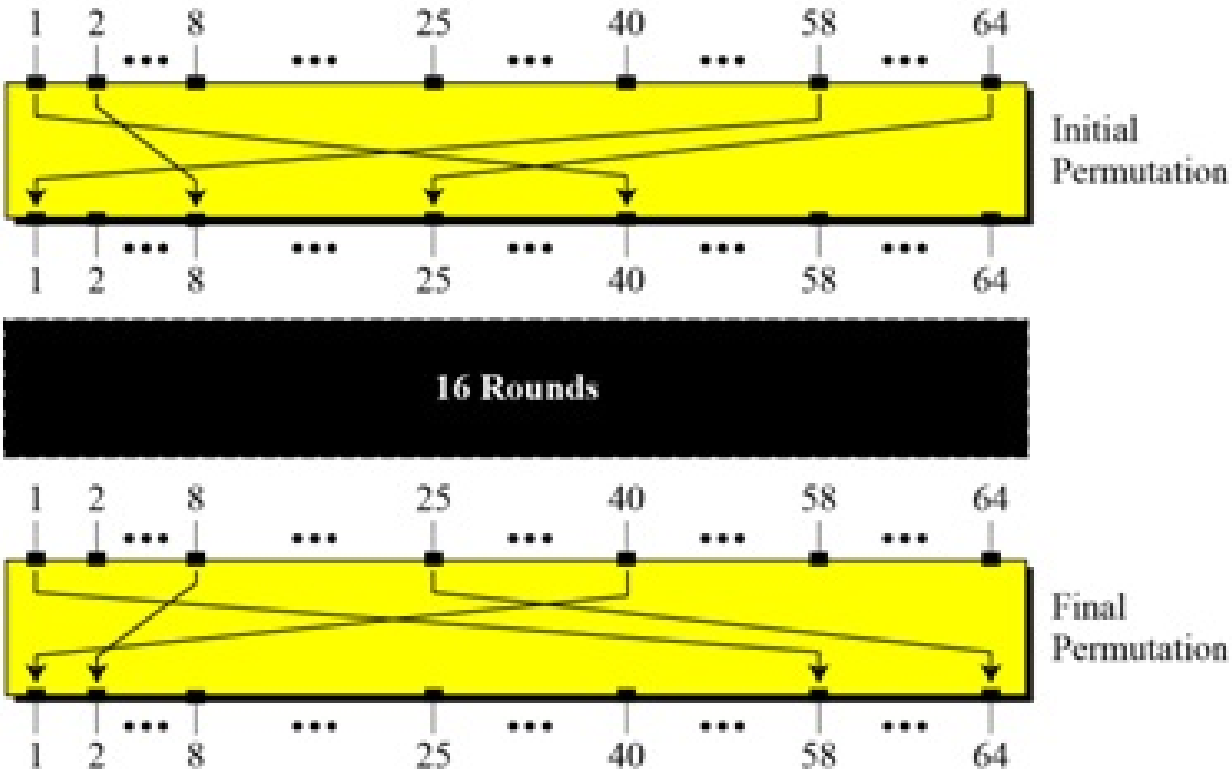
Adım-1

IP (Initial Permütasyon) başlangıç permütasyonu şifreleme işleminde girilen (plaintext) metine uygulanacak ilk permütasyondur. Burada bitler başlangıç permütasyonu tablosuna göre değiştirilip bir sonraki döngüye giriş olarak verilir. Aşağıda başlangıç ve bitiş Permütasyon tablosu verilmiştir.

Initial Permutation							
58	50	42	34	26	18	10	02
60	52	44	36	28	20	12	04
62	54	46	38	30	22	14	06
64	56	48	40	32	24	16	08
57	49	41	33	25	17	09	01
59	51	43	35	27	19	11	03
61	53	45	37	29	21	13	05
63	55	47	39	31	23	15	07

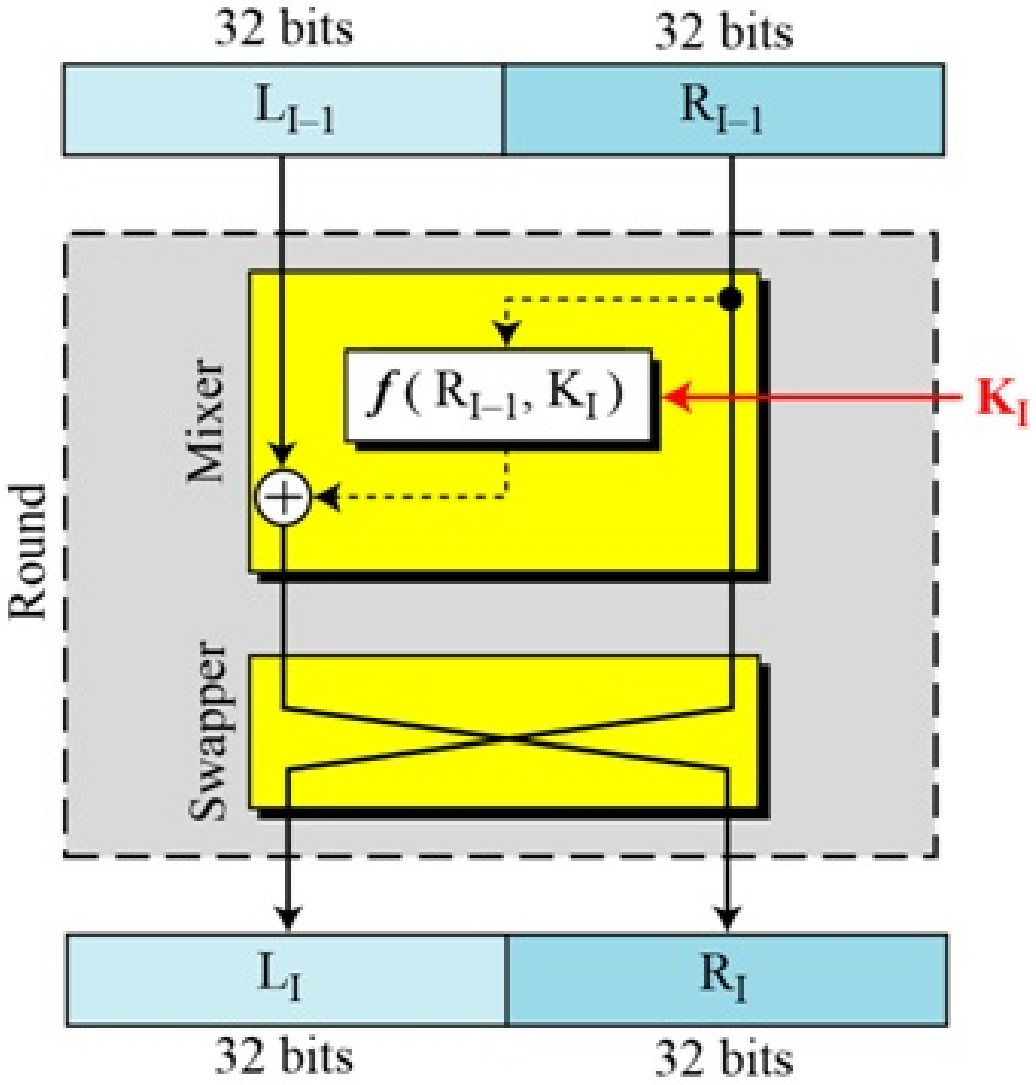
Final Permutation							
40	08	48	16	56	24	64	32
39	07	47	15	55	23	63	31
38	06	46	14	54	22	62	30
37	05	45	13	53	21	61	29
36	04	44	12	52	20	60	28
35	03	43	11	51	19	59	27
34	02	42	10	50	18	58	26
33	01	41	09	49	17	57	25

Başlangıç ve Bitiş Permutasyon tablosundaki değerler aşağıda verilen şemada gösterilen şekilde kullanılır.

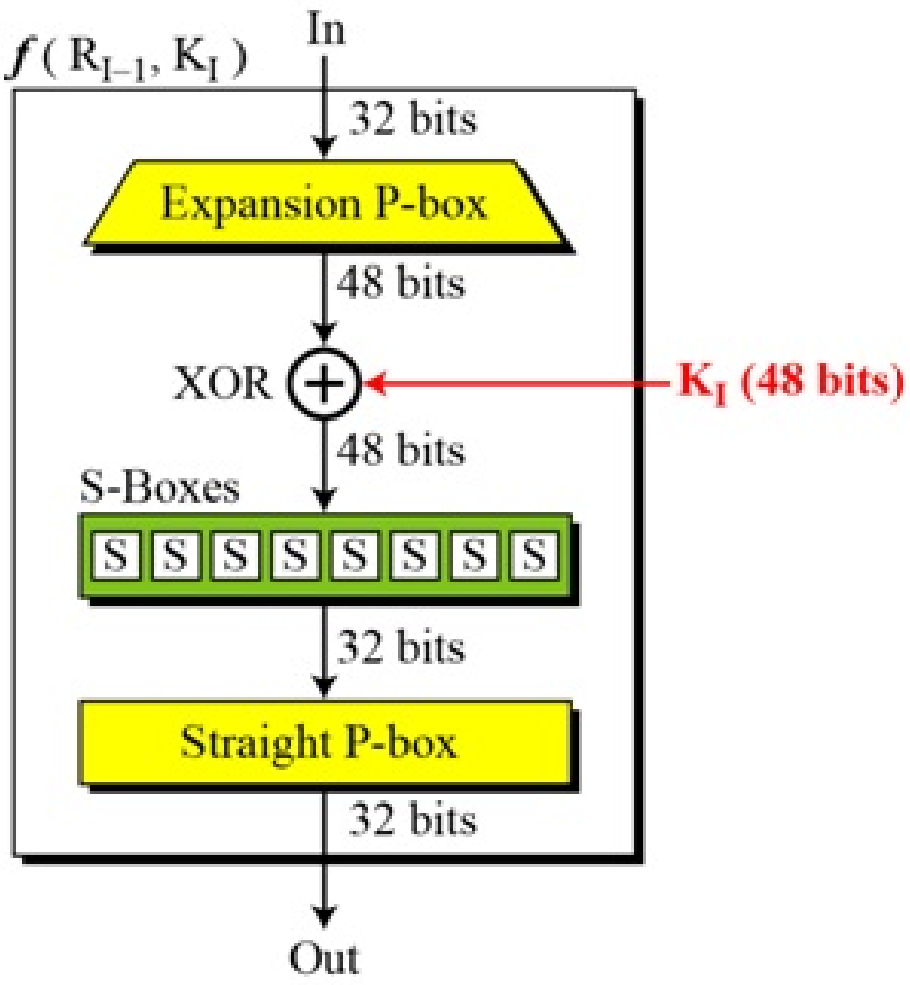


64-bitlik metin başlangıç permütasyonundan geçtikten sonra girişler döngü fonksiyonuna verilir. DES 16 kez döngü kullanır. Her döngü fiestel şifre dır, yani her adımda veri iki eşit uzunluktaki blok halinde şifrelenir.

DES ile veri güvenliğini sağlama yönteminde DES in önemli bir aşamasına gelindi. Bu aşamada ise 16 döngünün her birinin nasıl gerçekleştiği verilecektir. Aşağıda bunun tablosu verilmiştir.



Şemada ki f fonksiyonu kod da verilen DesFunction (DES fonksiyonu)dur. Burada 64-bitlik veri 32-bitlik sağ ve sol diye iki parçaya ayrılır. Sağ parçaya anahtar ile birlikte DesFunction uygulanır. Bu fonksiyonun çıkışı sol parça ile Xor lanarak bir sonraki turun sağ parçasını oluşturur. Aşağıda ki şema da DesFunction un şeması verilmiştir.



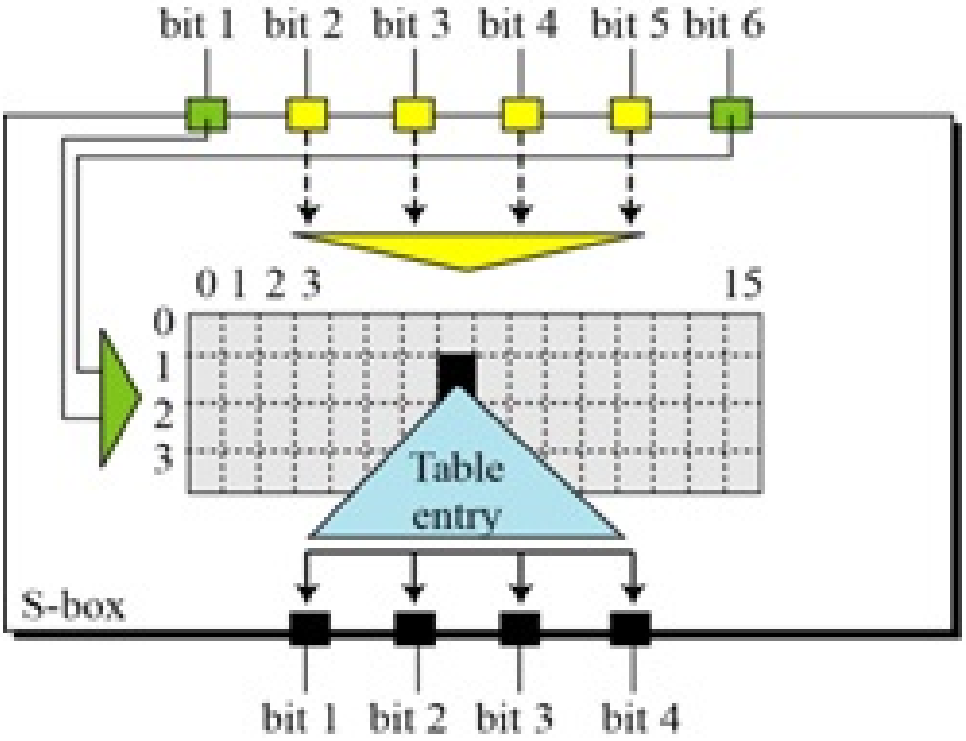
Şemada da görüldüğü gibi 32-bitlik var olan tur daki veri Expansion PBox Table yardımı ile 48-bite dönüştürülür. Bu 48-bitlik veri , 48

bitlik anahtar değeri ile Xor lanarak Sbox'a verilir.

Expansion PBox Table verilmiştir.

32	01	02	03	04	05
04	05	06	07	08	09
08	09	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	31	31	32	01

DES algoritması 8 tane 6-bitlik SBox kullanır. Bu SBox ların çıkışı 4-bittir. SBox'ın hangi girişine hangi çıkışın karşı geldiğini SBox tablosundan öğreniyoruz. Bu tablo aşağıda verilmiştir.



S1

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
1	00	15	07	04	14	02	13	10	03	06	12	11	09	05	03	08
2	04	01	14	08	13	06	02	11	15	12	09	07	03	10	05	00
3	15	12	08	02	04	09	01	07	05	11	03	14	10	00	06	13

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
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

S2

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
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

S3

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
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	9	4	5	11	12	7	2	14

S4

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	2	2	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	6	15	0	9	10	4	5	3

S5

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
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

S6

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S7

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
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

S8

S kutularından çıkış için bir örnek inceleyelim:

6 bitlik S1 kutusu girdisi 011011 olsun. Bu kutudan çıkan 4 bitlik veriyi bulmak istiyoruz. 1. ve 6. Bitler yani 0 ve 1 yan yana alınır 01 olur bunun karşılık gelen değeri 0+1=1 olduğundan S1 kutusundaki 1. Satır alınacaktır. 6 bitlik girdimizin geri kalan bitleri olan 1101 in karşılık gelen değeri ise 8+4+0+1=13 olduğundan S1 kutusundaki 13. Sütun alınır. Buna göre 1. Satır 13. Sütundaki değere bakılır, bu değer 5 dir. Bunun karşılığı ise 5=0101 olur. Son bulunan 4 bitlik ifade s1 kutusu girdisinin çıktısı olur.

8 adet 4-bitlik SBox çıkışları son olarak permutasyondan geçer. DES fonksiyonu içindeki StraightPBox fonksiyonu aşağıda verilen tablodaki şekilde bit değişimi yapar :

16	07	20	21	29	12	28	17
01	15	23	26	05	18	31	10
02	08	24	14	32	27	03	09
19	13	30	06	22	11	04	25

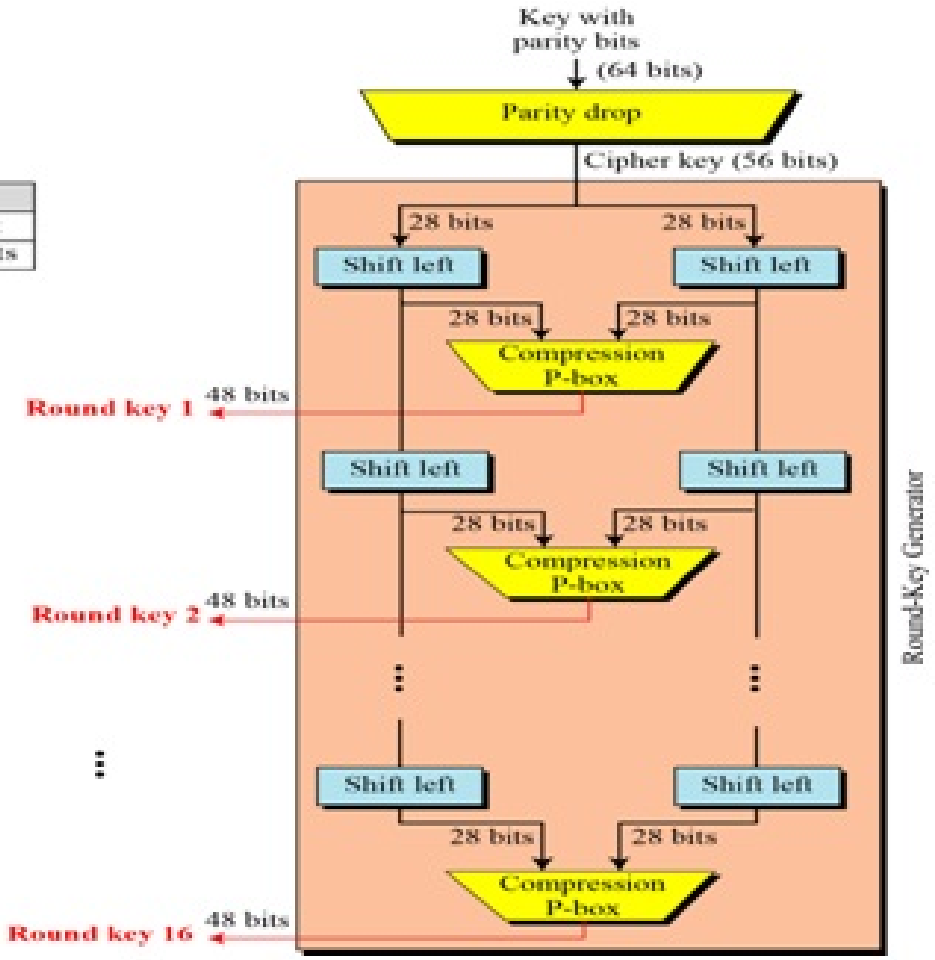
Son permütasyon sonundaki 32-bit DES fonksiyonunun çıkışı olur. Böylece DES algoritmasındaki tüm fonksiyonlar tamamlanarak 64-bitlik anahtar ile 64-bitlik düz metin, 64-bitlik şifreli metine dönüştürülmüş olur.

Döngü Anahtarı Üretimi

Yukarıda anlatılan kısımda 16 adet Döngü Anahtarının nasıl üretildiğinden bahsetmedik. Şimdi bu anahtarlar nasıl üretilir onu açıklayalım. Anahtar üretiminin işleyişi aşağıdaki resimde gösterilmiştir: Burada 64-bitlik değer 56-bite haritalanır. Parity drop tablosunu kullanarak bu işlem gerçekleştirilir. Bu tablo aşağıda verilmiştir:

Shifting

Rounds	Shift
1, 2, 9, 16	one bit
Others	two bits



57	49	41	33	25	17	09	01
58	50	42	34	26	18	10	02
59	51	43	35	27	19	11	03
60	52	44	36	63	55	47	39
31	23	15	07	62	54	46	38
30	22	14	06	61	53	45	37
29	21	13	05	28	20	12	04

Bu 56-bitlik şifre anahtarı ikiye bölünerek 28 bitlik anahtarlar elde edilir. Bu değerler shift left (sola kaydırma) yapılarak çıkışlar CompressionPBox'a verilir. 1,2,9 ve 16 roundlarında 1 bit diğerlerinde 2 bit kaydırma yapılır.

Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bit shifts	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Kaydırma circular left (kendi üzerinden dönme) şeklinde yapılır.

Kaydırma işleminden sonra CompressionPbox'a gelen çıkışlar 48-bit olarak çıkar. Bu 48-bitlik değer o anki döngünün anahtar değeri olur.

CompressionPboxTable ve dizisi aşağıda verilmiştir:

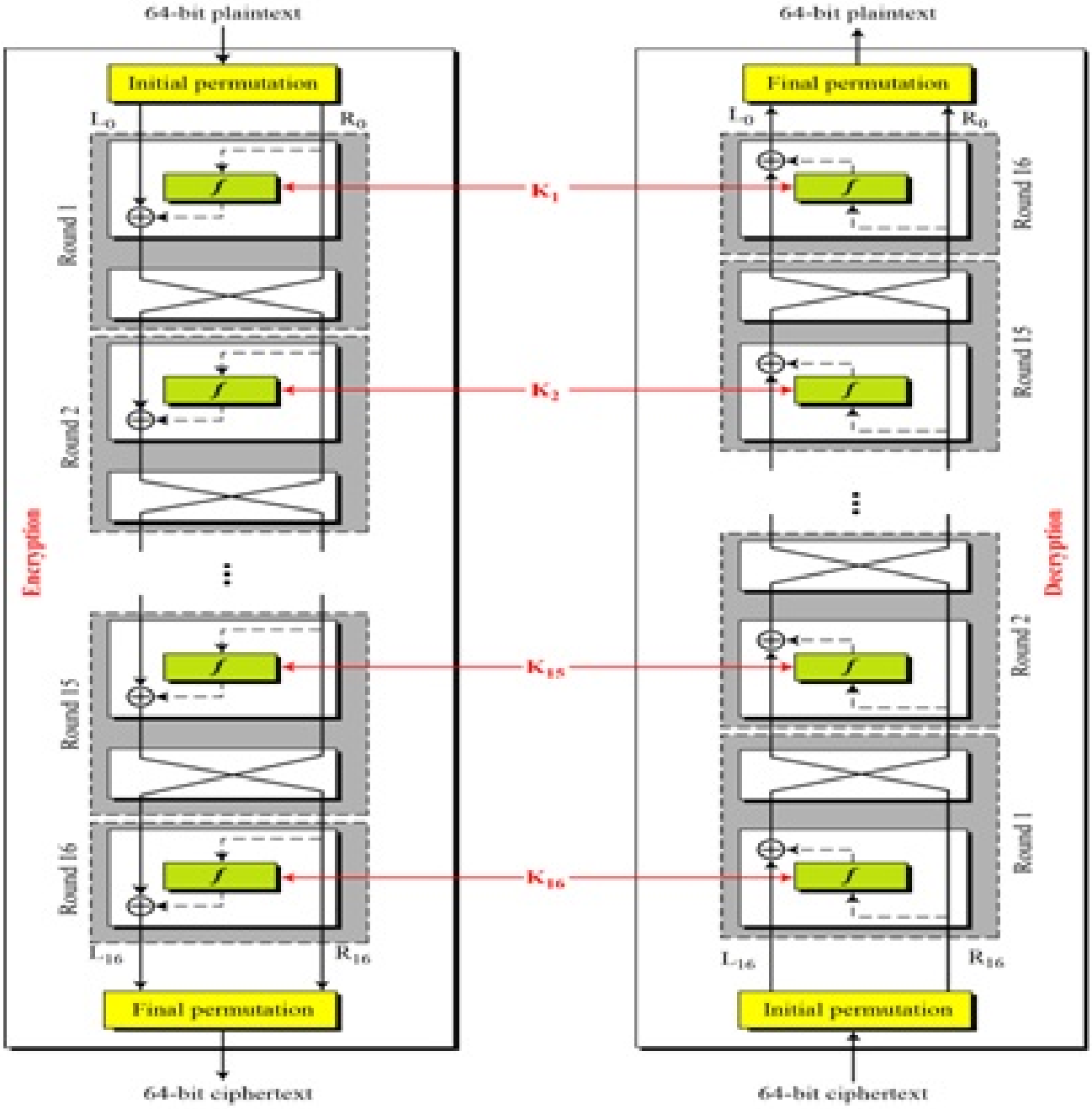
14	17	11	24	01	05	03	28
15	06	21	10	23	19	12	04
26	08	16	07	27	20	13	02
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

Deşifreleme İşlemi

Des algoritmasında mesajı şifreledikten sonra şifreli mesaj aynı şekilde deşifrelenir. Tek fark şifrelenirken her adımda verilen anahtar burada tersten verilir. Yani deşifrelenirken ilk adımda anahtar 16.anahtar ikinci adımda 15.anahtar şeklindedir.

Bu sıra 1.anahtara doğru bir şekilde yukarıdan aşağıya doğru gitmektedir. Aşağıda verilen

şekilde şifreleme ve deşifreleme işlemi gösterilmiştir. Bu şekilden de görüldüğü gibi şifreleme ile deşifreleme arasındaki tek fark tersi yönde kullandıkları anahtarlardır.


64-bit plaintext
Initial permutation
L_0
R_0
Round 1
Round 2
Round 15
Round 16
Encryption
K_1
K_2
K_{15}
K_{16}
L_{16}
R_{16}
Final permutation
64-bit ciphertext
64-bit plaintext
Final permutation
L_0
R_0
Round 16
Round 15
Round 2
Round 1
Decryption
L_{16}
R_{16}
Initial permutation
64-bit ciphertext