BÖLÜM 3

DES

3.1 DES Algoritması

DES (Data Encryption Standard) algoritması, 1970 yılında IBM tarafından geliştirilen Lucifer algoritmasının biraz daha geliştirilmiş halidir. 1974'te IBM'in NSA ile birlikte geliştirdiği algoritma olan DES'in yayınlanmasından itibaren DES algoritması üzerinde geniş ölçüde çalışmalar yapılmıştır.

İlk tasarladığında donanım uygulamalarında kullanılması amaçlanmıştır. İletişim amaçlı kullanımda hem gönderen, hem de alıcı şifreleme ve deşifrelemede kullanılan aynı gizli anahtar üzerinde anlaşmış olmalıdır. Gizli anahtarın güvenli bir biçimde dağıtımı için açık anahtarlı sistem kullanılabilir. DES aynı zamanda sabit diskte veri saklamak gibi tek kullanıcılı şifreleme amaçlı da kullanılabilir. DES'in en büyük zayıflığı 56 bitlik anahtarıdır. Geliştirildiği zamanlarda çok iyi bir şifreleme algoritması olmasına rağmen modern bilgisayarlar tarafından yapılan anahtar saldırılarına karşı yetersiz kalmıştır. DES'in diğer bir zayıflığı da yavaş olmasıdır.

DES algoritması Feistel yapısındadır. DES'i 16 döngüden oluşan bir döngüye benzetebiliriz. İlk döngüye girmeden önce başlangıç permütasyonu ve son döngüden sonra da başlangıç permütasyonunun tersi uygulanır. DES algoritmasını aşağıdaki şekilde gösterilebilir.

DES Algoritması

Her döngü bir önceki döngüden gelen mesajı ikiye ayırır: L_i ve $R_i, i = 1, 2, \ldots, 16$. İşlemler R_i üzerinde yapılır. Her döngü için anahtardan döngü anahtarları üretilir. Deşifreleme işleminde de aynı algoritma kullanılır. Ancak anahtarların kullanım sırası tersten olur. Şimdi algoritmanın bileşenlerinin tek tek inceleyelim.

3.1.1 Başlangıç Permütasyonu

Başlangıç permütasyonu DES'e hiçbir kuvvet katmamaktadır.Başlangıç permütasyonunu aşağıda verilmiştir.

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

Görüldüğü üzere, başlangıç permütasyonunda 58. bit 1. bit yerine, 50.bit 2. bit yerine,... gelmektedir.

3.1.2 Başlangıç Permütasyonun Tersi

Başlangıç permütasyonunun tersi olan permütasyon son rounddan sonra uygulanır.Bu permütasyon aşağıda verilmiştir.

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

3.1.3 Anahtar Permütasyonu ve Döngü Anahtarının Üretilmesi

Anahtar üzerine ilk işlem 64 bitten 56 bite indirgemektir.Bunun için her 8. bit doğruluk kontrolü(parity check)için atılır. Daha sonra 56 bitlik anahtar aşağıda verilen permütasyona girer.

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 permütasyondan sonra 56 bitlik anahtar 28 bitlik sağ ve sol olmak üzere iki parçaya ayrılır. Döndürme(rotation) olarak adlandırdığımız kısımda, 28 bitlik parçalar her döngü için 1 yada 2 bit sola kayar.Bu kaydırma döndürme olarak adlandırılır çünkü kayan bitler sona eklenir.

Döngü	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
Kayma Miktarı	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Daha sonra anahtarı döngüye göndermeden önce tekrar bir permütasyon gerçekleşir. Bu permütasyon sonucu 56 bit 48 bite iner. Bu permütasyon 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

3.1.4 f fonksiyonu

Önceden de bahsedildiği üzere her döngüde sağ 32 bitlik kısım (R_i) üzerine işlemler yapılır. Öncelikle bu 32 bitlik kısım aşağıdaki gibi 48 bite genişletilir.

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 30 31 32 01

48 bitlik bu kısım döngü anahtarı ile x-or işlemine ($\oplus$) gönderilir. Sonra 48 bit, 6 bitlik 8 gruba bölünür ve her bir grup ayrı bir S-kutusuna gönderilir. S-kutularında 6 bitler 4 bite çevrilir. 8 S-kutusu aşağıda verilmiştir.

1. S-kutusu

	00	01	02	03	04	05	06	07	08	09	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	01	10	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

Örnek 3.1.1 *girdi = 101110 satır = 10(ilk ve son bitler) = 2 , sütun = 0111(ortada kalan bitler) = 7 çıktı = 11 (onbir) = 1011*

2. S-kutusu

	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
0	15	01	08	14	06	11	03	04	09	07	02	13	12	00	05	10
1	03	13	04	07	15	02	08	14	12	00	01	10	06	09	11	05
2	00	14	07	11	10	04	13	01	05	08	12	06	09	03	02	15
3	13	08	10	01	03	15	04	02	11	06	07	12	00	05	14	09

3. S-kutusu

	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
0	10	00	09	14	06	03	15	05	01	13	12	07	11	04	02	08
1	13	07	00	09	03	04	06	10	02	08	05	14	12	11	15	01
2	13	06	04	09	08	15	03	00	11	01	02	12	05	10	14	07
3	01	10	13	00	06	09	08	07	04	15	14	03	11	05	02	12

4. S-kutusu

	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
0	07	13	14	03	00	06	09	10	01	02	08	05	11	12	04	15
1	13	08	11	05	06	15	00	03	04	07	02	12	01	10	14	09
2	10	06	09	00	12	11	07	13	15	01	03	14	05	02	08	04
3	03	15	00	06	10	01	13	08	09	04	05	11	12	07	02	14

5. S-kutusu

	00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15
0	02 12 04 01 07 10 11 06 08 05 03 15 13 00 14 09
1	14 11 02 12 04 07 13 01 05 00 15 10 03 09 08 06
2	04 02 01 11 10 13 07 08 15 09 12 05 06 03 00 14
3	11 08 12 07 01 14 02 13 06 15 00 09 10 04 05 03

6. S-kutusu

	00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15
0	12 01 10 15 09 02 06 08 00 13 03 04 14 07 05 11
1	10 15 04 02 07 12 09 05 06 01 13 14 00 11 03 08
2	09 14 15 05 02 08 12 03 07 00 04 10 01 13 11 06
3	04 03 02 12 09 05 15 10 11 14 01 07 06 00 08 13

7. S-kutusu

	00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15
0	04 11 02 14 15 00 08 13 03 12 09 07 05 10 06 01
1	13 00 11 07 04 09 01 10 14 03 05 12 02 15 08 06
2	01 04 11 13 12 03 07 14 10 15 06 08 00 05 09 02
3	06 11 13 08 01 04 10 07 09 05 00 15 14 02 03 12

8. S-kutusu

	00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15
0	13 02 08 04 06 15 11 01 10 09 03 14 05 00 12 07
1	01 15 13 08 10 03 07 04 12 05 06 11 00 14 09 02
2	07 11 04 01 09 12 14 02 00 06 10 13 15 03 05 08
3	02 01 14 07 04 10 08 13 15 12 09 00 03 05 06 11

S-kutuları blok şifrelerin doğrusal olmayan kısımlarıdır. Hiçbir S-kutusu girdinin doğrusal yada afin fonksiyonu değildir.

S-kutularından çıkan 4 bitlik parçalar yanyana gelerek birleşir ve aşağıdaki permütasyona gider.

16 07 20 21

29 12 28 17

01 15 23 26

05 18 31 10

02 08 24 14

32 27 03 09

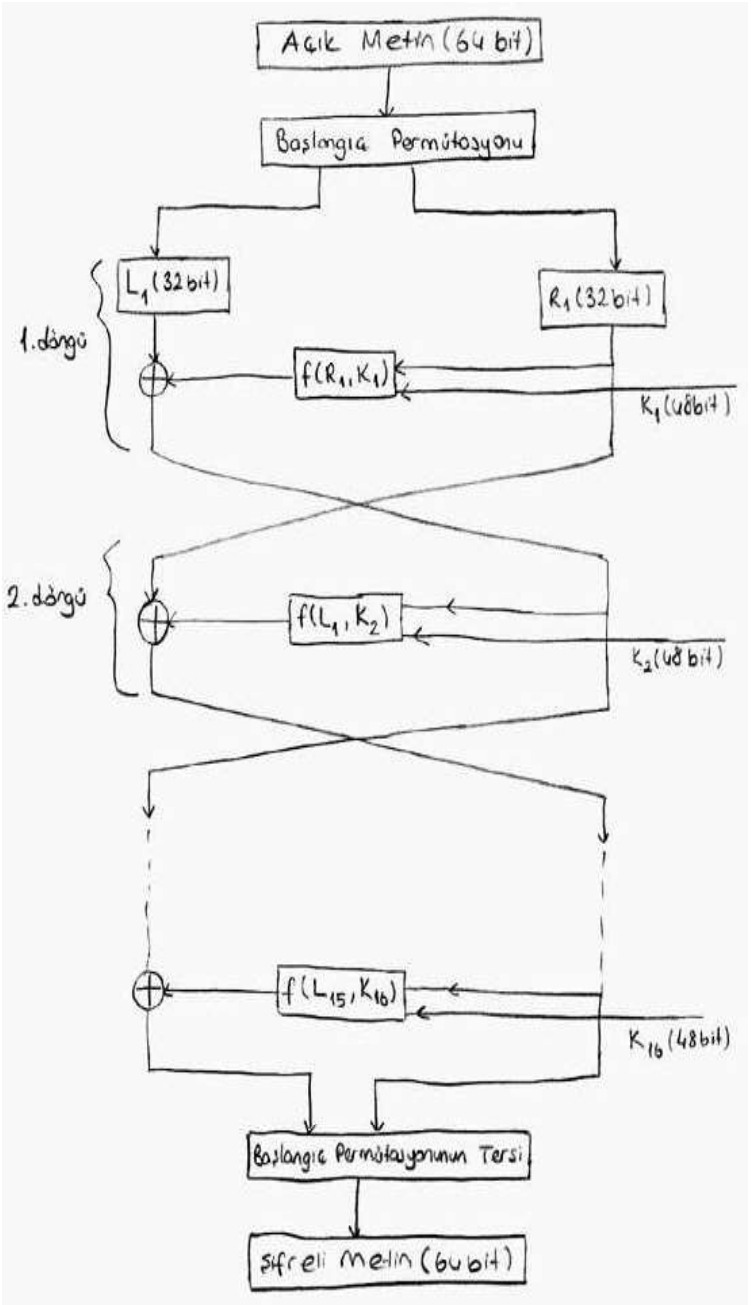
19 13 30 06

22 11 04 25

Permütasyondan çıkan 32 bitlik kısım döngünün başaında ayrılan 32 bitlik kısımla XOR işlemi uygulanır.

3.2 DES'in Tasarım Özellikleri

DES'in en önemli özelliği yayılma(confusion) ve nüfuz etme(diffusion)özellikleridir. DES'te her bloğun her biti diğer bitlere ve anahtarın her bitine bağımlıdır. Bunun iki amacı vardır:Öncelikle, anahtar üzerindeki bilinmezlik(uncertanity)artmaktadır. İkinci olaraksa, açık metindeki veya anahtardaki 1 bitin değişmesi bütün şifreli metinin değişmesine sebep olur ki bu bizim ileride öğreneceğimiz diffrensiyel kriptanalizde işe yaramaktadır.


Açık Metin (64 bit)
Başlangıç Permütasyonu
L_1 (32 bit)
R_1 (32 bit)
1. döngü
$f(R_1, K_1)$
K_1 (48 bit)
2. döngü
$f(L_1, K_2)$
K_2 (48 bit)
$f(L_{15}, K_{16})$
K_{16} (48 bit)
Başlangıç Permütasyonunun Tersi
Şifreli metin (64 bit)