

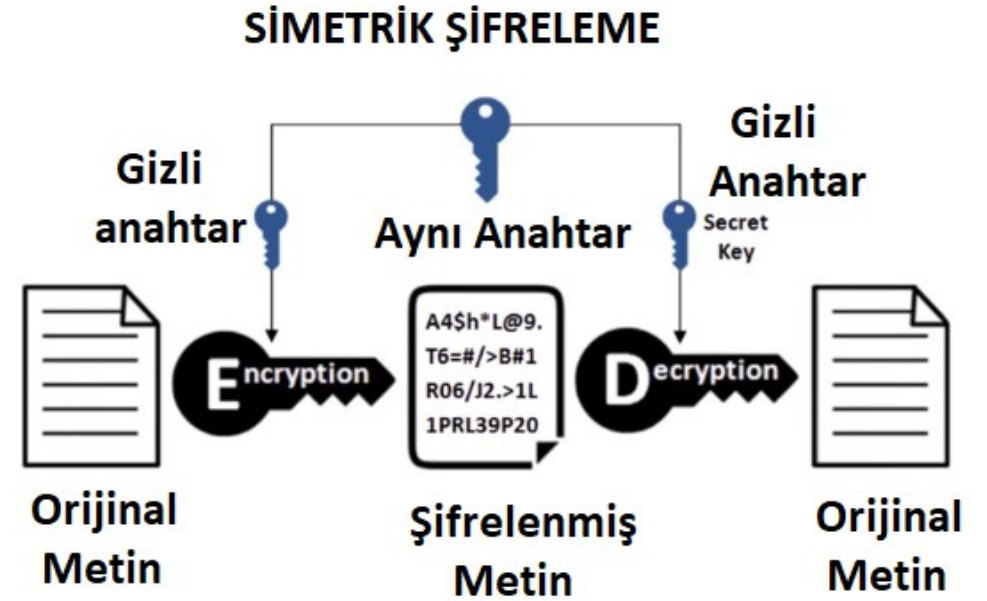
# BİLGİ GÜVENLİĞİ DERSİ

## 6. HAFTA

SİMETRİK VE ASİMETRİK ALGORİTMALAR

# SİMETRİK ŞİFRELEME

- Simetrik şifreleme tek anahtara dayanan şifreleme türüdür.
- Veriyi şifrelemede kullanılan anahtar çözümleme amacıyla da kullanılır.
- Veriler bloklar halinde işlenir.



# SİMETRİK ŞİFRELEME

- Simetrik şifrelemede karakterlerin yerine bit kullanılır (a->b yerine a->00010110)

# SİMETRİK ŞİFRELEME

Veriler blok halinde şifrelendiği için blok şifreleme tabanlı bir sistemdir. Bu nedenle blok şifrelemenin 2 önemli esasını barındırır:

**Karışıklık (Confusion):** anahtar ile orijinal metnin arasındaki ilişkinin gizlenmesidir. (a harfine karşılık gelen anahtar değerine bakarak b harfinin değerini tahmin etme)

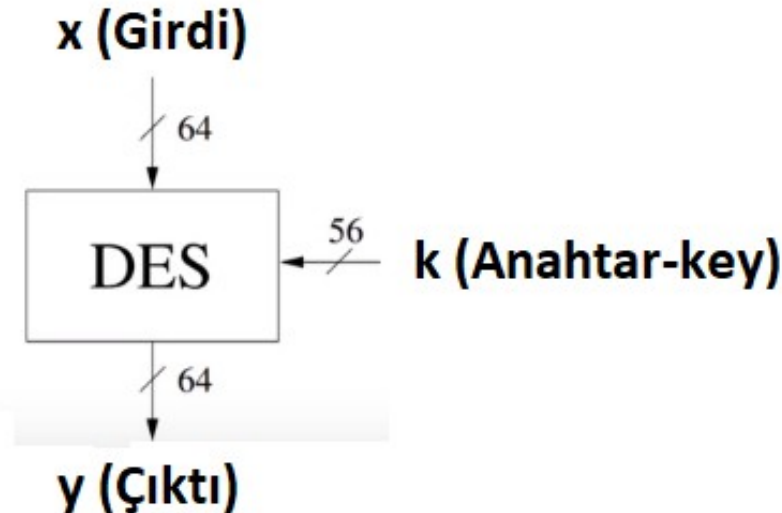
**Yayılma (Diffusion):** Orijinal metnin istatistiklerini şifreli metnin daha geniş bir alanına yayarak gizlenmesini sağlar («Merhaba» yazısı için 128 bitlik bir hash değerinin üretilmesi gibi)

# DES (Data Encryption Standard) Algoritması

- DES (Data Encryption Standard) ilk simetrik şifreleme algoritmasıdır.
- IBM tarafından 1970'lerde geliştirilmiş ve NSA tarafından 1977'de bir standart haline gelmiştir.
- DES, bir blok şifreleme örneğidir.
- Yıllar içinde üzerinde çeşitli kriptolojik ataklar denenmiş, anahtar uzunluğunun kısa olması nedeniyle yeterli güvenliği sağlayamadığı görülmüştür.
- Daha sonradan yerini Triple DES ve AES algoritmalarına bırakmıştır.

# DES Algoritmasının Yapısı

- Algoritmadaki matematiksel işlemler bitler (0 ve 1) üzerinden gerçekleştirilir.
- DES algoritmasının blok uzunluğu 64 bit, anahtar uzunluğu ise 56 bittir.
- Şifrelenecek olan veriye açık metin (plain text) denir. Bu metin 64 bitlik bloklara ayrılır, her blok birbirinden bağımsız olarak şifrelenir ve aynı uzunlukta şifrelenmiş metinler (cipher text) elde edilir.



# DES Algoritmasının Yapısı-Anahtar Uzunluğu

Normalde anahtar uzunluğu da 64 bittir. Ancak, 8 bit parite bit olarak ayrılır. Parite bit, her 7 bitin yanına eklenen bir kontrol bitidir. Bu bit, 7 bit içerisinde tek sayıda 1 biti varsa 1 olarak eklenir ve o diziyi çift yapar. Çift sayıda bit varsa 0 olarak eklenir. Bu durum, bitlerin kontrolü için kullanılır.

# DES Algoritmasının Yapısı

Şifreleme işlemi yapılırken kullanılan mantıksal işlem XOR'dur. XOR, farklı bitler için 1 çıktısını üretirken (0-1 ->1 , 1-0 ->1), aynı bitler için 0 çıktısını üretir (0-0 ->0 , 1-1 ->0). Bit seviyesinde işlemler

*complement*

```
~ 01010001110101110000000000001111
   1010111000101000111111111110000
```

*bitwise and*

```
01010001110101110000000000001111
& 00110001011011100011000101101110
   00010001010001100000000000001110
```

*bitwise or*

```
01010001110101110000000000001111
| 00110001011011100011000101101110
   01110001111111111001100010110111
```

*bitwise xor*

```
01010001110101110000000000001111
^ 00110001011011100011000101101110
   01100000101110010011000101100001
```



# DES Algoritmasının Yapısı

DES Algoritması genel itibari ile 3 adımdan oluşur.

- 1) Başlangıç Permütasyonu işlemi (Initial Permutation - IP)
- 2) 16 defa tekrar eden şifreleme işlemi (16 Round)
- 3) Tersine Permütasyon işlemi (Inverse Permutation)

# DES Algoritması-Başlangıç Permütasyonu

Veriler 64 bitlik bloklar haline getirildikten sonra bu blokların her birine 16 Round'luk şifreleme işlemi uygulanır. 64 bitlik bloklara yerleştirilen veriler, sıralı düzenden çıkarılır ve permütasyon tablosundaki indeksleme sırasına göre

- Başlangıç permütasyonu, DES algoritmasının ilk aşamasıdır. Dışarıdan gelen 64 bitlik açık metindeki bitler initial permutation tablosundaki koordinatlara göre yer değiştirirler.

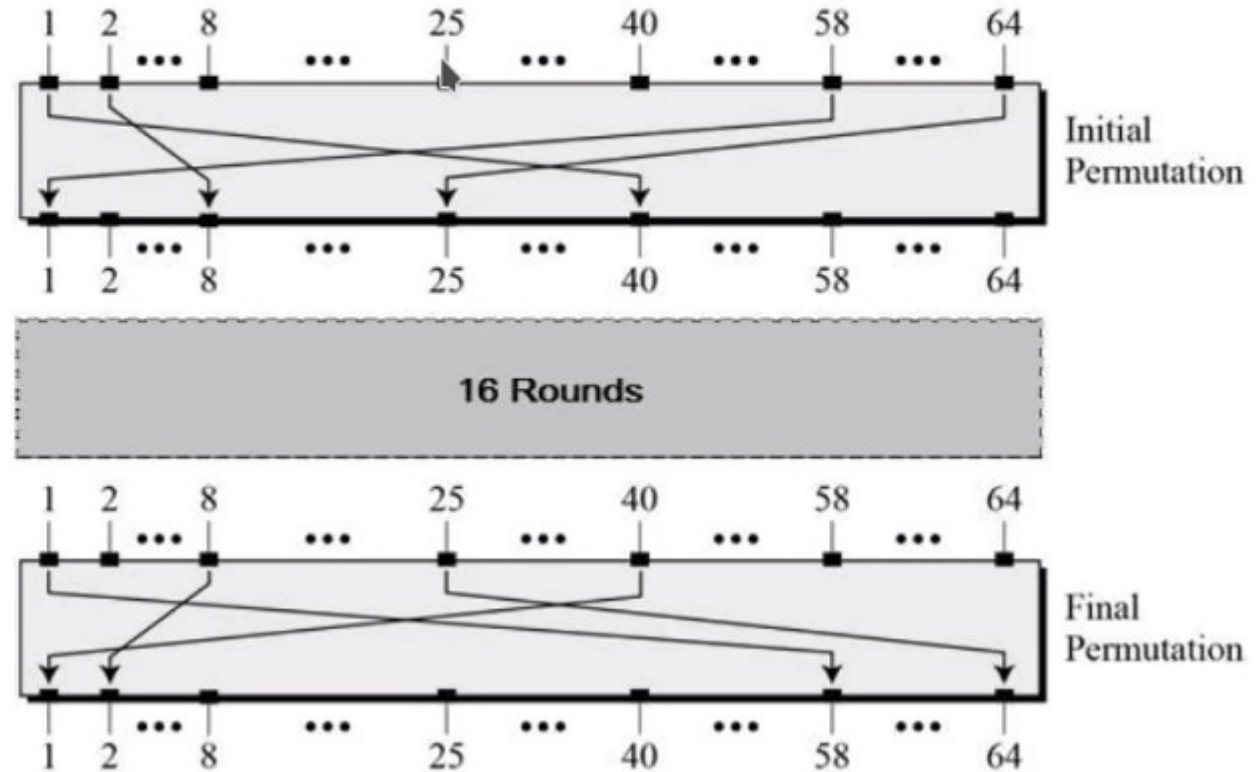
## IP

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

# DES Algoritması-Başlangıç Permütasyonu

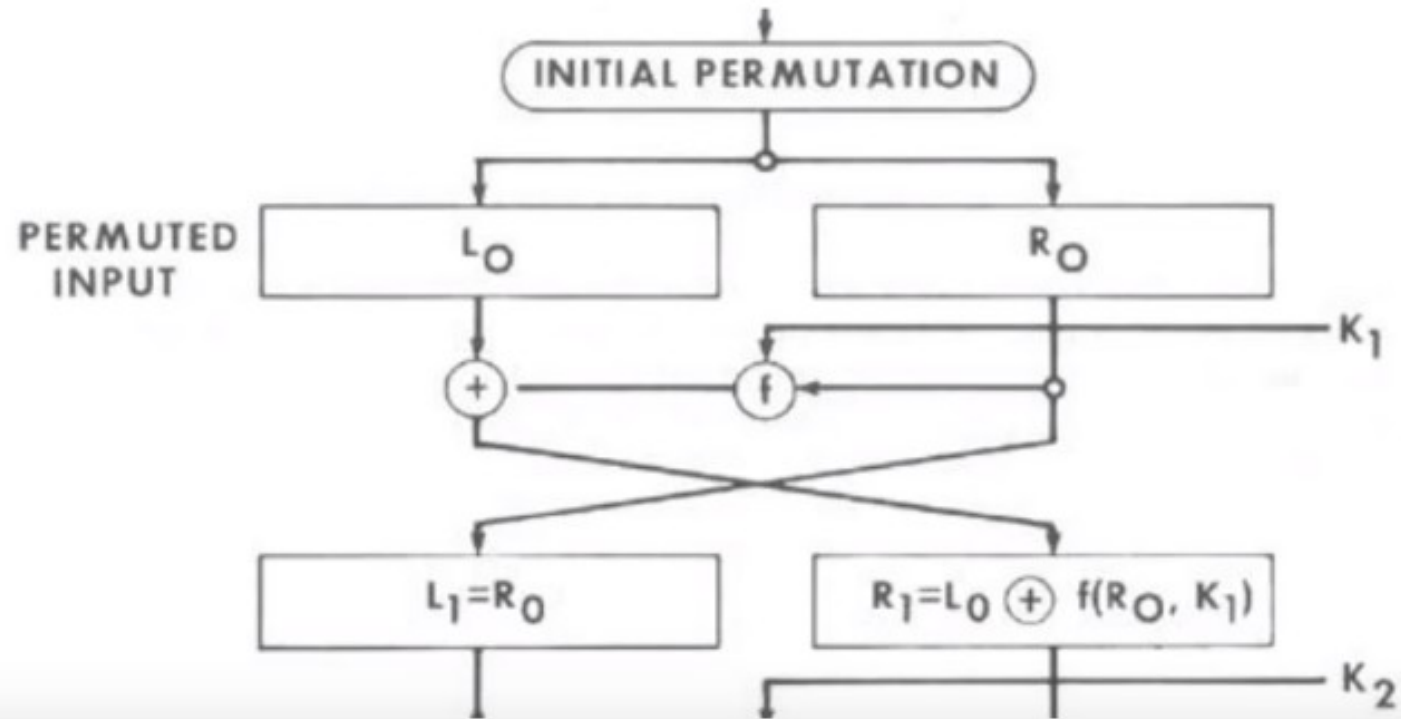
Aşağıdaki şekilde görüldüğü gibi permütasyon işlemi sonrası 1. bit 40. bitin yerine, 2. bit 8. bitin yerine, 58. bit 1. bitin yerine yazılmıştır-yerini almıştır.

- Permütasyon işlemine tabi tutulan 64 bitlik veri döngü aşamasına iletilir.



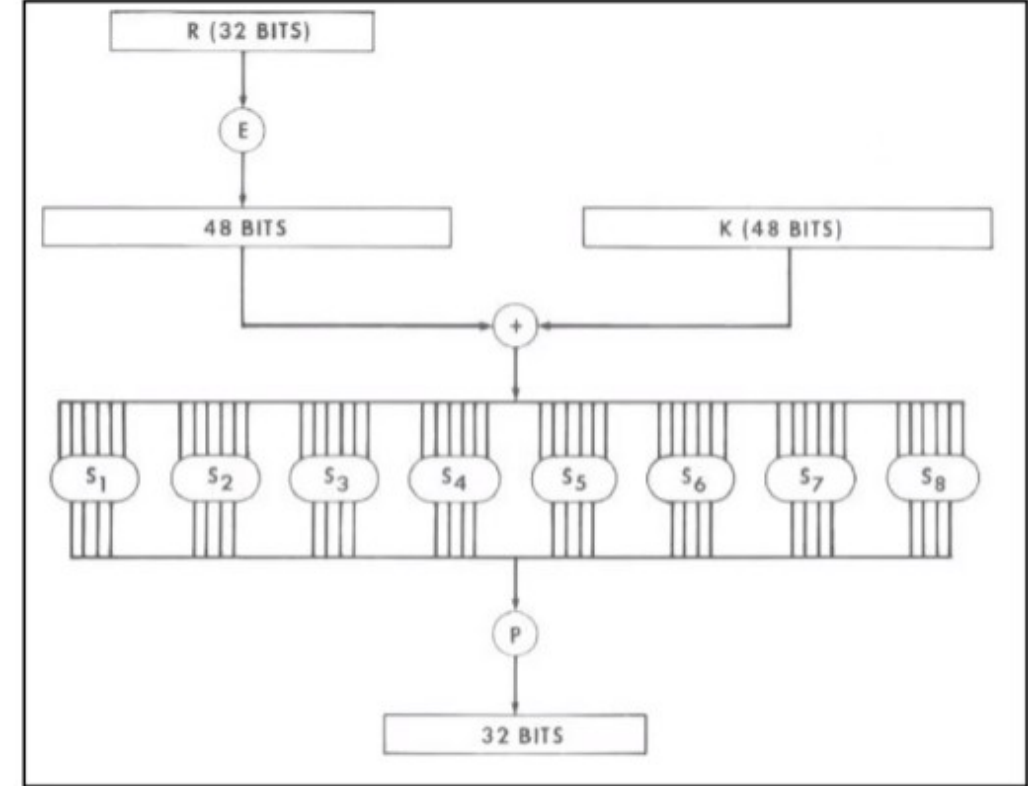
# DES Algoritması-DES Döngüsü (Round)

- DES döngüsü iki ana işlemten oluşur. Bunlar F fonksiyonu ve XOR işlemleridir.
- Başlangıç permütasyonundan gelen veri sağ ve sol şeklinde 32 bitlik iki parçaya ayrılır.



# DES Algoritması- F Fonksiyonu

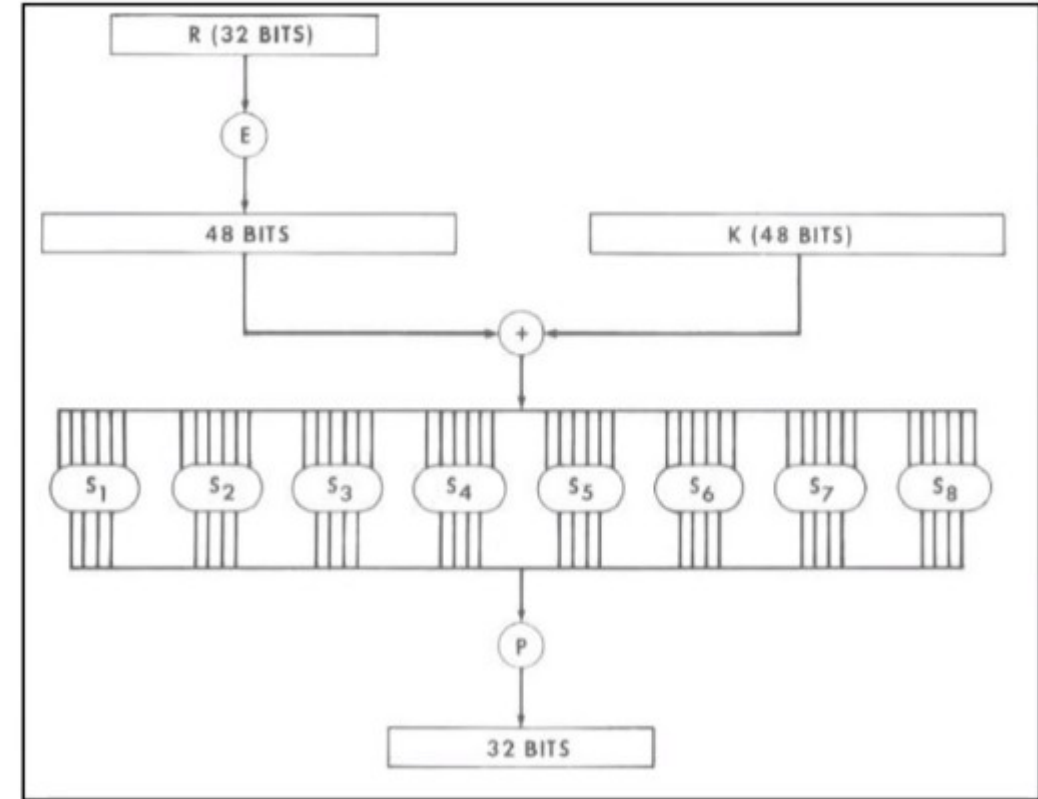
- F fonksiyonuna giren sağ 32 bitlik veriye ilk olarak expansion (genişletme) işlemi uygulanır. Bu işlemde 32 bitlik verinin bazı bitleri tekrarlanarak 48 bitlik veri elde edilir.
- Anahtar üretici tarafından, 56 bitlik anahtar girişinden her döngü adımı için ayrı olarak kullanılacak 16 adet 48 bitlik anahtar üretilir.



# DES Algoritması- F Fonksiyonu

S kutusu (S box), verilerin işlendikten sonra yerleştirildiği bloklardır. 8 adet S kutusu vardır. Bu kutuların girişi 6 bit ( **$6 \times 8 = 48$  bit**), çıkışı ise 4 bittir ( **$4 \times 8 = 32$  bit – başlangıçtaki bit sayısı**)

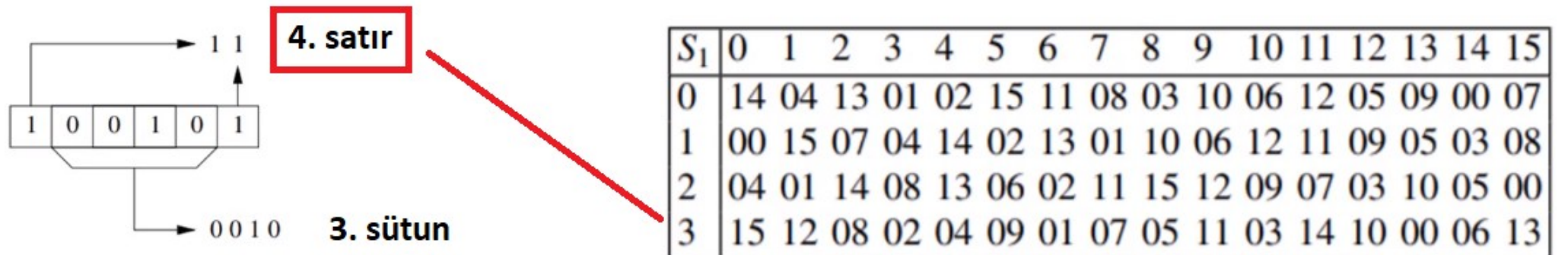
- Genişletme işleminden gelen veri ve anahtara XOR işlemi uygulanır.
- Çıkan veri S kutularına girer.



# DES Algoritması- S Kutusu

S kutusunun girişindeki 6 bitin ilk ve son biti (şekildeki 1 ve 1), yan yana dizilir ve ikili tabandaki karşılığına bakılır.  $0011 = 0+0+2.1+1.1 = 2+1=3$ . Bu işlemin sonucu olan sayının S kutusundaki satır indeksine gidilir (4. satır).

- F fonksiyonun içinde 8 adet S kutusu bulunur. Bu kutuların girişi 6 bit çıkışı ise 4 bittir.
- Her S kutusu 4 satır 16 sütundan oluşur ve 64 adet 4 bitlik sayıyı barındırır.
- Örnek :  $S_1(37 = 100101) = 8 = 1000$

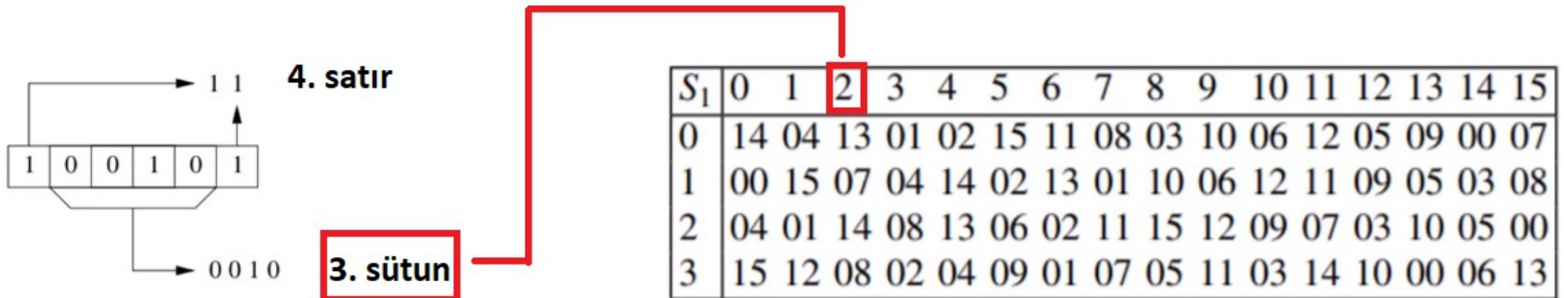




# DES Algoritması- S Kutusu

Daha sonra ortadaki 4 bit yan yana yazılır ve ikili sistemden onluk sisteme çevrilir. Örneğimize göre, 0010  $\rightarrow 8.0+4.0+2.1+1.0 = 2$ . Bu işlemden çıkan sonuca ait sütun indeksine gidilir (2 değeri, 3. sütunda yer alıyor).

- F fonksiyonun içinde 8 adet S kutusu bulunur. Bu kutuların girişi 6 bit çıkışı ise 4 bittir.
- Her S kutusu 4 satır 16 sütundan oluşur ve 64 adet 4 bitlik sayıyı barındırır.
- Örnek :  $S_1(37 = 100101) = 8 = 1000$

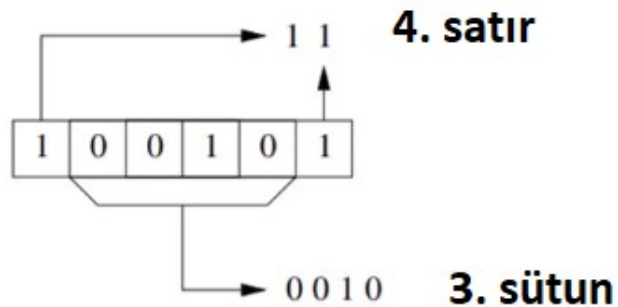




# DES Algoritması- S Kutusu

Son olarak, 4. Satır (3 numaralı indeks) ve 3. sütunun (2 numaralı indeks) kesişimi olan değer, girdi sayısı olan 37'nin yeni değeri olur. Yani 37'nin yeni değeri 8 olur (ikili sistemde 1000).

- F fonksiyonun içinde 8 adet S kutusu bulunur. Bu kutuların girişi 6 bit çıkışı ise 4 bittir.
- Her S kutusu 4 satır 16 sütundan oluşur ve 64 adet 4 bitlik sayıyı barındırır.
- Örnek :  $S_1(37 = 100101) = 8 = 1000$



$S_1$	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	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

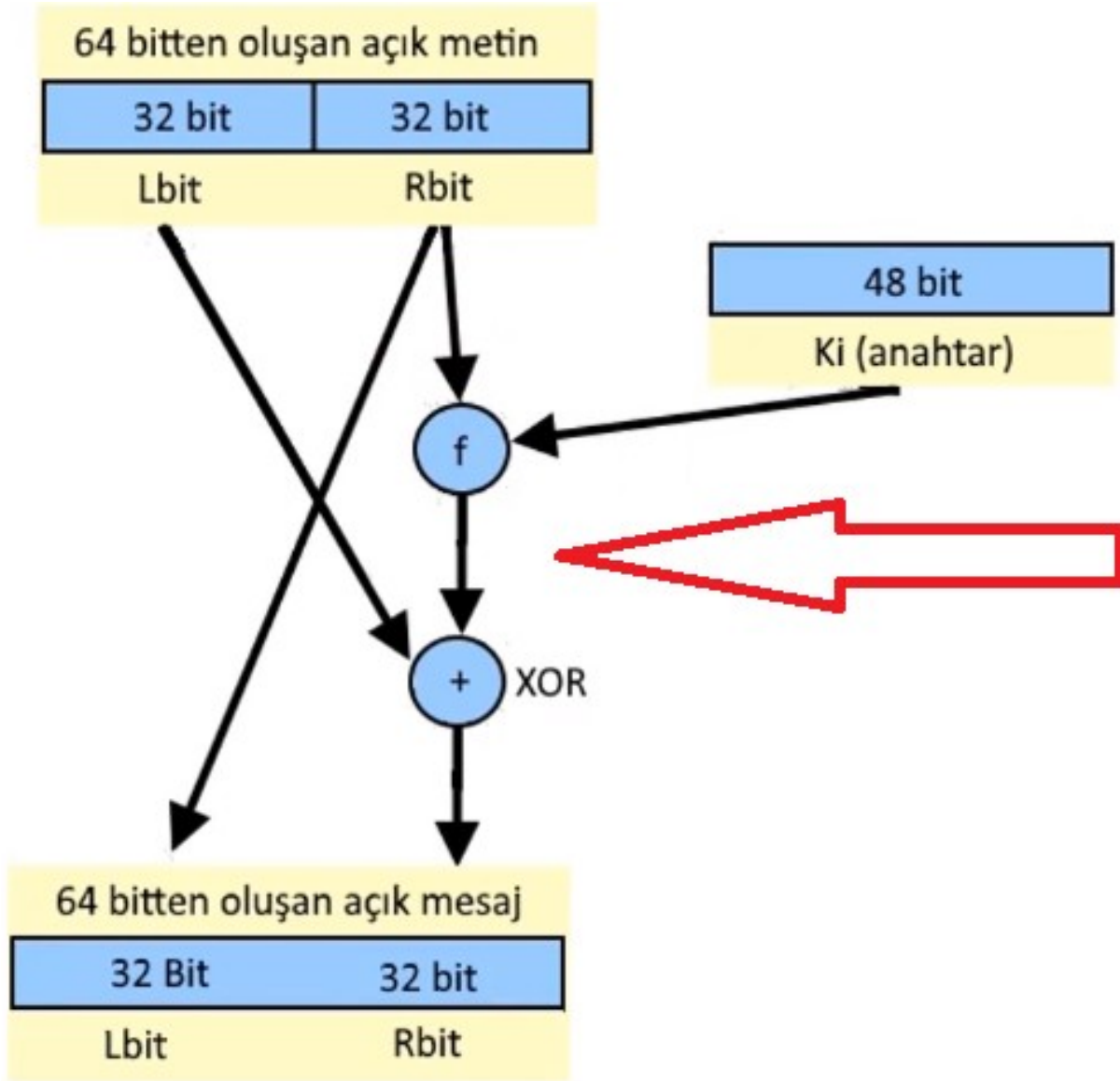
# DES Algoritması- F Fonksiyonu

S kutusu ile orijinal bit sayısı (başlangıçtaki bit sayısı) elde edildikten sonra bu bitlere de tekrar bir permütasyon işlemi daha uygulanıyor ve f fonksiyonu tamamlar. Sekiz adet S kutusunun çıkışından toplam 32 bitlik veri elde edilir.

- Elde edilen veriye permütasyon işlemi uygulanır.

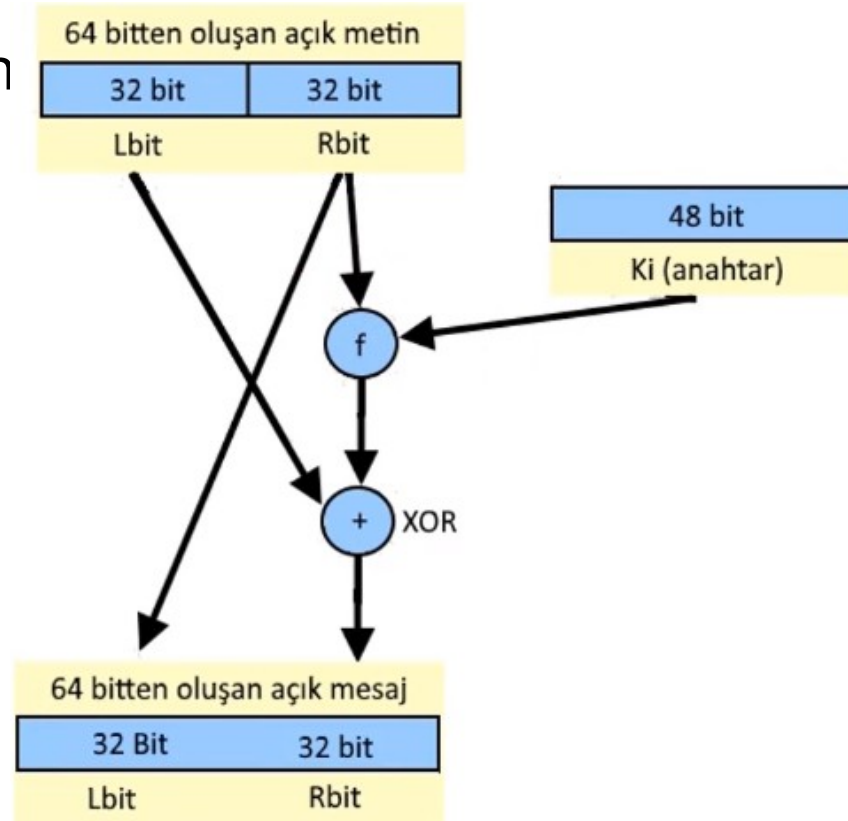
$P$							
16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

# DES Algoritması- XOR İşlemi



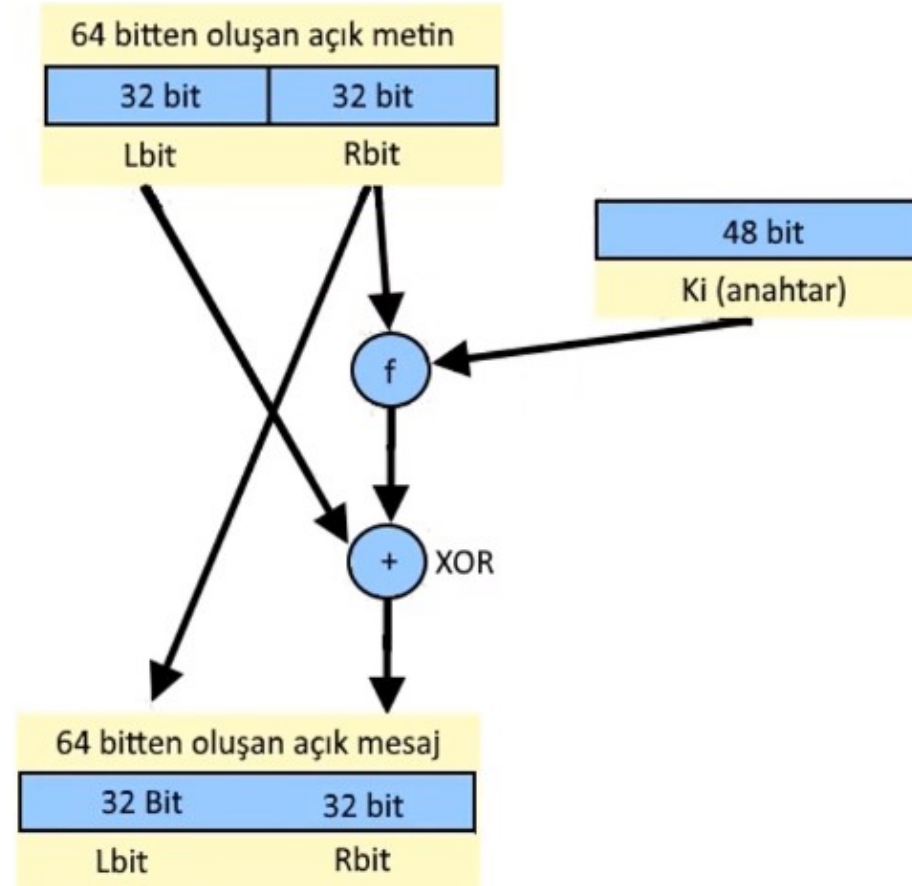
# DES Algoritması- XOR İşlemi

Şu ana kadar yaptığımız tüm işlemleri, en başta 2 gruba ayırdığımız 32 bitlik verilerden sağ taraftaki (R) 32 bitlik veriye uygulandığını düşünün. XOR işlemi, bu aşamalardan geçmiş sağ taraftaki 32 bitlik veri ile hiçbir işlemde geçmemiş sol taraftaki 32 bitlik veri arasında uygulanır ve oluşan yeni veri bir sonraki adımın (Round'un



# DES Algoritması- XOR İşlemi

2. Adımda ise aynı işlemler sol taraftaki 32 bitlik veriye uygulanacak ve sağ taraftaki veri ile XOR işlemine tâbi tutulacak. 16 Round boyunca tekrar eden bu işlemler ile sağ taraftaki veriler 8 defa, sol taraftaki veriler de 8 defa şifrelenecektir.



# DES Algoritması- Ters Permütasyon İşlemi

Ters Permütasyon, DES Algoritmasının son adımıdır. Başlangıç permütasyonundaki işlem tersten uygulanır ve 64 bitlik şifrelenmiş metin (cipher text) çıkışa verilir.

$IP^{-1}$

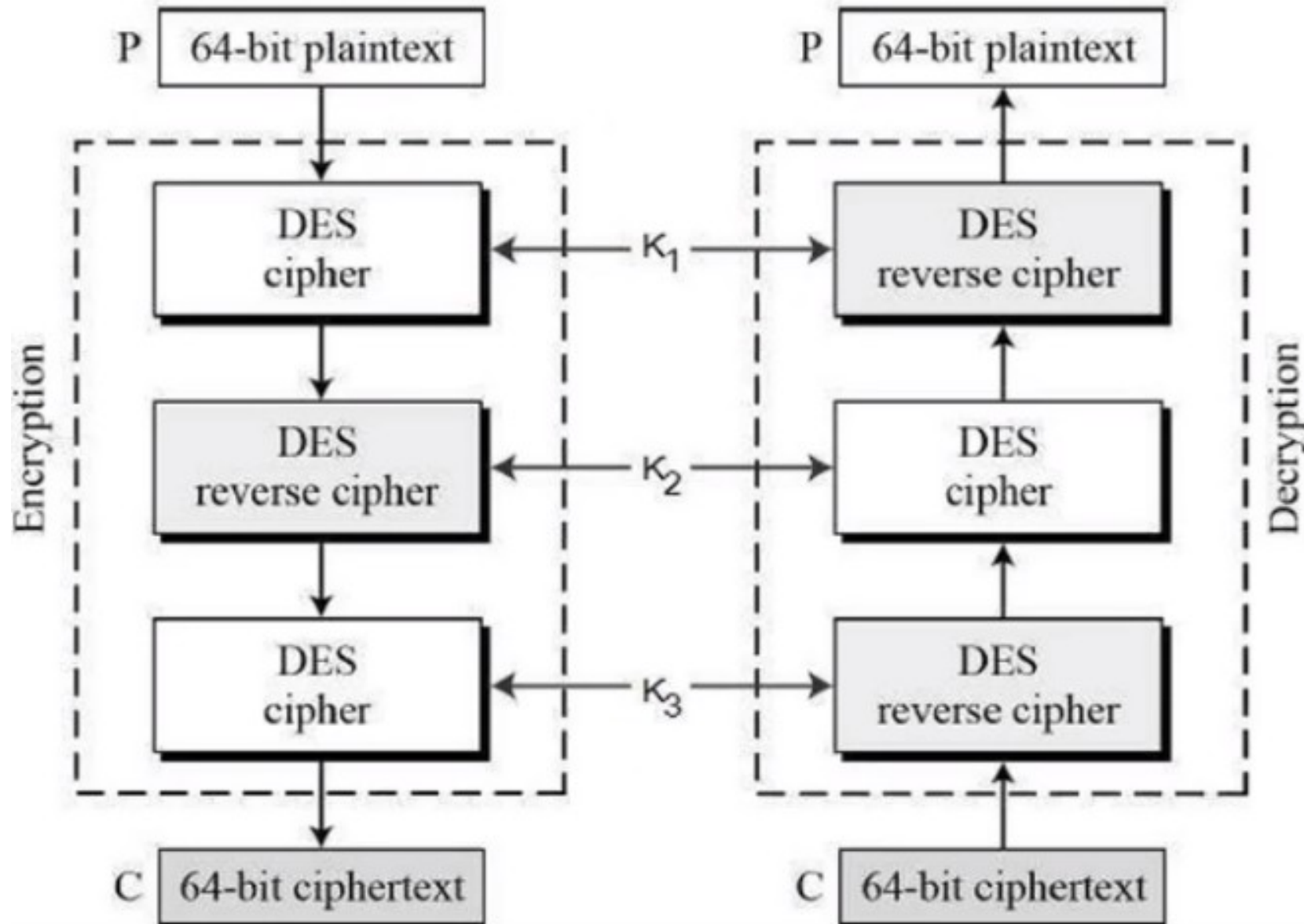
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25



# DES Algoritmasının Güvenliği

- Bilgisayar teknolojisinin gelişmesiyle şifreleme algoritmalarına saldırılar kolaylaşmıştır.
- DES algoritmasının 56 bitlik anahtar uzunluğu ile bu saldırılara karşı yeterli güvenliği sağlayamadığı görülmüştür.
- Brute-force ataklarına dayanıklı değildir.
- DES algoritmasının daha zor saldırılır hale gelmesi için **168** bit anahtar uzunluğu kullanan üçlü DES uygulaması geliştirilmiştir.
- 3DES bankalar ve devlet daireleri başta olmak üzere birçok ortamda kullanılmaya devam etmektedir

# Triple DES (3DES) Algoritmasının Yapısı





# ASİMETRİK ŞİFRELEME

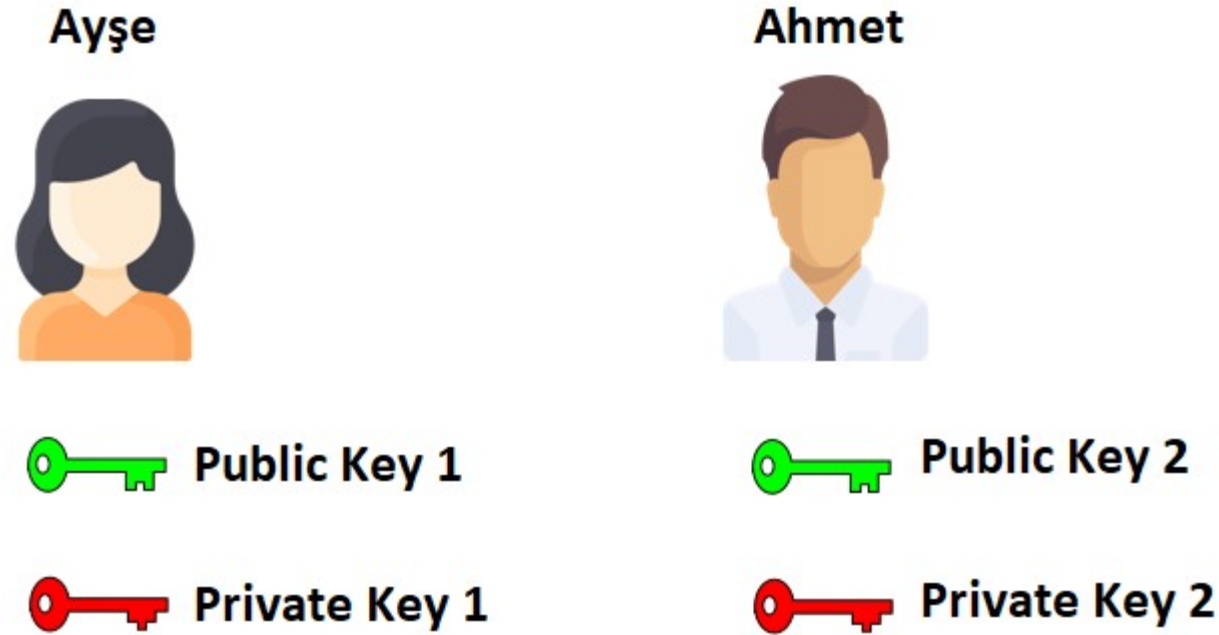
Simetrik şifrelemede, veriyi şifrelemek ve şifre çözmek için aynı anahtar kullanılıyordu. Bu senaryoda, anahtarın ele geçirilmesi durumunda mesaj da çözülebiliyordu.

# ASİMETRİK ŞİFRELEME

Asimetrik şifrelemede ise iki kişi arasındaki güvenli oturumda 4 anahtar kullanılır.

Herkesin, bir açık anahtarı (public key), bir de kişisel-özel anahtarı (private key) bulunur.

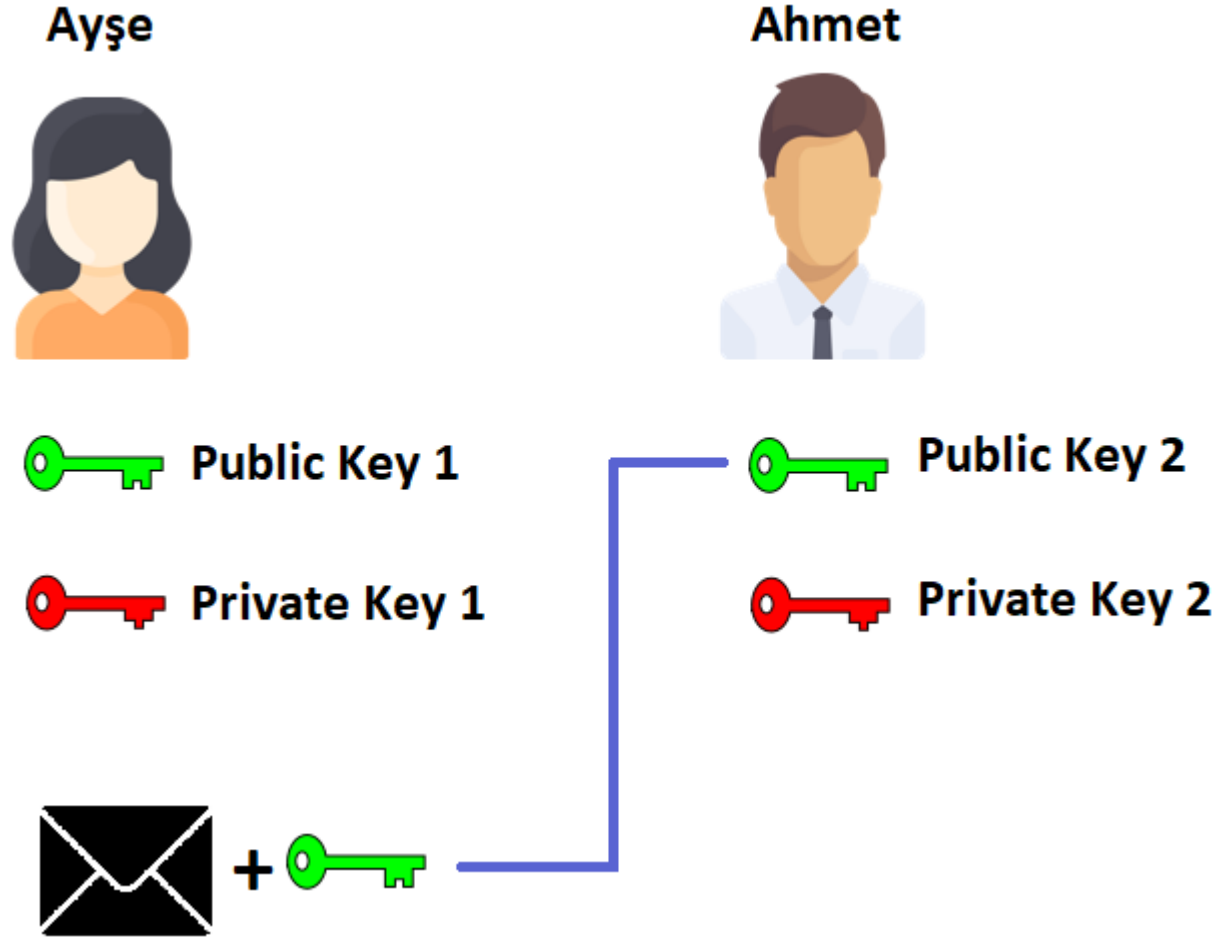
# ASİMETRİK ŞİFRELEME



- Şekilde görüldüğü gibi, 2 kişinin asimetrik şifreleme ile mesaj paylaşımında bulunduklarını düşünelim.
- Her iki kullanıcının da kullandıkları 2 anahtar bulunmaktadır

# ASİMETRİK ŞİFRELEME

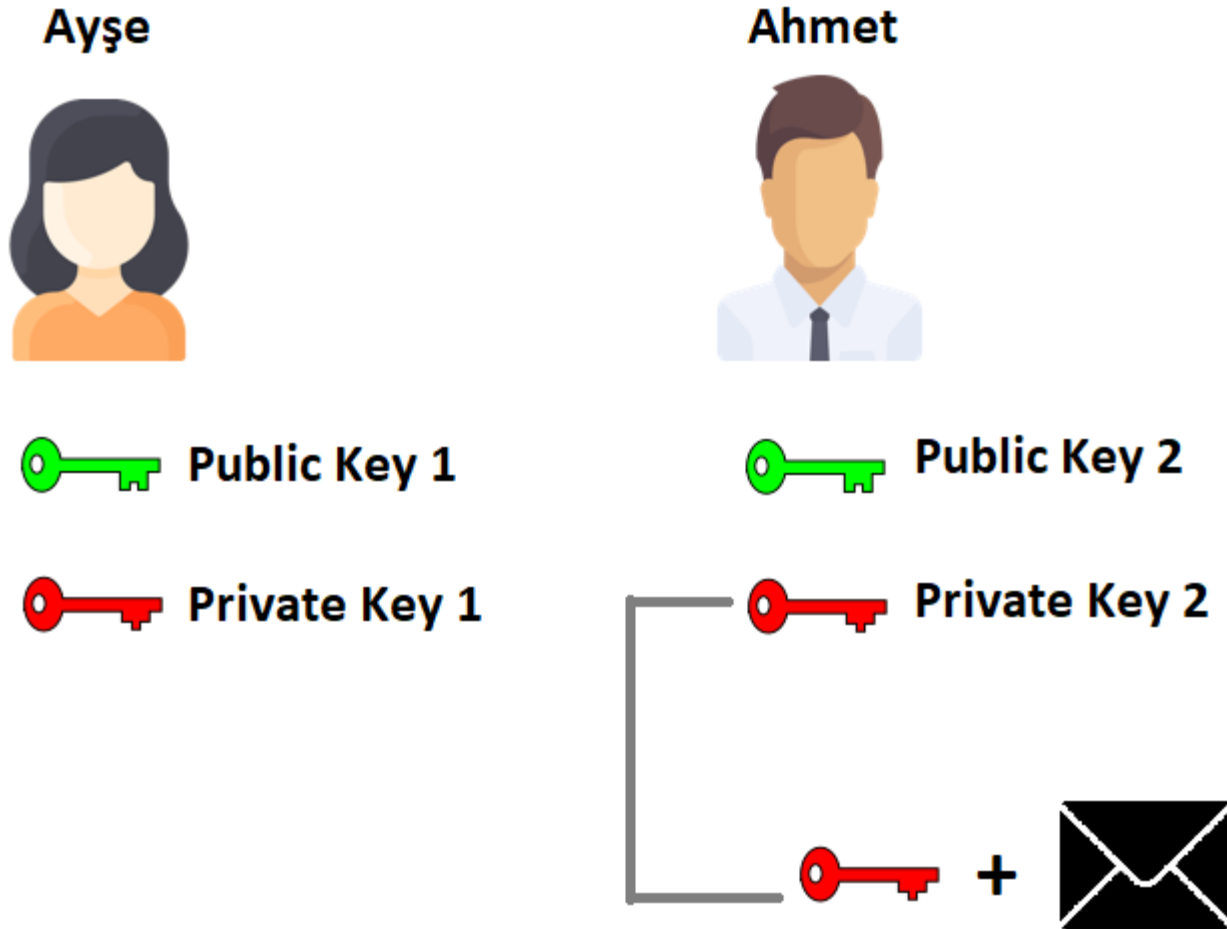
Ayşe, Ahmet'e bir mesaj göndereceği zaman bu mesajı şifrelemesini yaparken Ahmet'in açık anahtarını kullanır (Public Key 2).



# ASİMETRİK ŞİFRELEME

Mesaj şifrelendikten sonra Ahmet'e iletilir.

Ahmet, mesajın şifresini çözerken kendi özel anahtarını kullanır (Private Key 2).



# ASİMETRİK ŞİFRELEME

**NOT:** Public key ve private key bilgileri, rassal sayı üreteçleri tarafından üretilir.

Kullanılan matematiksel işlemler ile private key bilgisi ile public key bilgisi arasında matematiksel bir ilişki vardır.

**ANCAK**, bu ilişki, private key'den public key'e göre **TEK YÖNLÜDÜR**.

# ASİMETRİK ŞİFRELEME

Peki, asimetrik şifrelemede kimlik doğrulama işlemi nasıl yapılır?

Yani, bir mesaj alındı ve şifresi çözüldü ama gönderen kişinin kim olduğu nasıl anlaşılır?

# ASİMETRİK ŞİFRELEME

Bu sorunun cevabı, «dijital imza»da yatmaktadır.

Mesajın sonuna, mesajın hash'i (özet bilgisi) gönderen kişinin private key'i ile şifrelenir.

Alıcı taraf, gönderenin public key bilgisine sahip olduğu için, bu şifreli mesajın doğruluğunu anlamak için şifrelenmiş özeti ortaya çıkarır.

Orijinal hash değeri ile karşılaştırır ve özet değerleri aynı ise, kimlik doğrulama işlemi sağlanmış olur.



# ASİMETRİK ŞİFRELEME

**UNUTMAYINIZ**, private key bilgisi ile public key bilgisi arasında matematiksel bir işlem vardı ve bu TEK YÖNLÜ bir işlemdi.

# RSA Algoritması

- RSA: Ron Rivest, Adi Shamir, Leonard Adleman
- $N = p * q$  , p ve q asal sayılar
- p ve q'nun çarpımını bulmak kolay
- Fakat N'den p ve q'yu tekrar elde etmek çok zor.
- N 1024, 2048, 4096-bit olabilir.



# KAYNAKLAR

Mehmet Akif AKKAYA