

## III. Gizli Anahtar Kriptografi

### III.I Simetrik Şifreleme

Kriptografi kullanıcılarının alet çantalarında şu altı araç bulunur:

- Simetrik şifreleme
- Hash fonksiyonları
- Mesaj kimlik doğrulama kodları
- Rastgele sayı jeneratörleri
- Genel anahtar şifreleme
- Dijital imzalar

Bu konuda ilk dördünü, bir sonraki konuda ise son ikisini göreceğiz.

### Simetrik Şifreleme

Genel anahtar şifrelemeye karşın, simetrik şifrelemede hem şifreleme hem de şifre çözme işlemlerinde aynı anahtar kullanılır. Simetrik bir şifre yapmanın iki yolu vardır:

- Akış şifresi: Şifreleme kuralı düzmetin sembollerinde sembolün yerine göre değişir. Vigenere, RC4, A5.
- Blok şifre: Birden fazla düzmetin sembolü tek bloğa şifrelenir. örn.:DES, AES, Twofish, RC6.

Blok şifrelerin örnekleri olarak DES ve AES'i kısaca tanıtacağız.

### DES'in tanımı

DES (Data Encryption Standard) 64 karakter uzunlukta bir 'x' düzmetin karakter dizisini 56 karakterli bir diziye şifreler. Algoritma üç basamaklıdır.

- Başlangıç Permütasyonu (IP):  $x_0 = IP(x) = L_0 R_0$
- Daha sonra belli bir fonksiyonun 16 tekrarı hesaplanır.
- Ters Permütasyon ( $IP^{-1}$ ):  $y = IP^{-1}(R_{16}L_{16})$

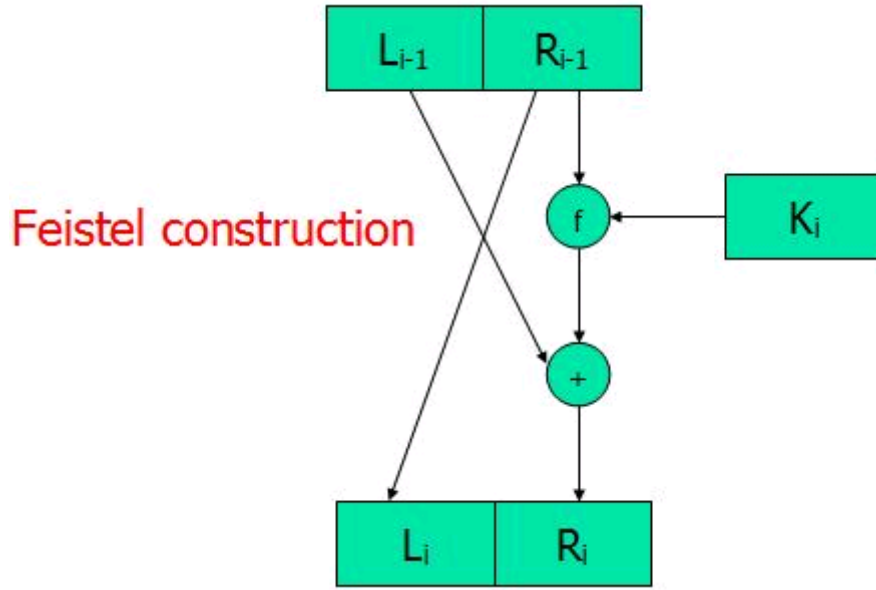
DES şifrelemenin bir döngüsü (Feistel yapısı da denir) sağdaki resimde gösterilmektedir. DES'in çalışmasındaki ana sorun "f fonksiyonunu" belirlemektir. Fonksiyon şu basamaklardan oluşur:

1. Genişleme fonksiyonu  $E(A)$  girdi  $A$ 'dan 32 bit içerir, bu bitler belli bir permütasyonla aynı 16 bitin iki kere gösterimidir.
2.  $B = E(A) \oplus J$  hesaplanır ( $J$  anahtar planı ile anahtardan üretilen 48 bit uzunluğundaki ikinci argümandır).
3. 8 S-box kullanılarak  $C_j = S_j(B_j)$  hesaplanır. ( $j$  1 ve 8 arası değerler alır)
4. Sabit permütasyon  $P$  uygulanır.

Anahtar planında, 56-bitlik anahtar onaltı adet döngü başına 48-bitlik anahtar yaratmada kullanılır, her anahtar için 56-bitin farklı bir 48-bitlik alt kümesini alır. S-box her 6-bit girdi için 4-bitlik bir çıktı üretir. Gördüğümüz üzere, geçen konuda gördüğümüz permütasyon ve ikame fikirleri, günümüzün modern şifrelerinde de kullanılır.

Şifrelemenin nasıl çalıştığının genel özeti budur. Şifre çözme temelde DES'i tersten çalıştırmayla olur.

DES'in basitliğini gören biri, herhangi birinin rahatlıkla bir şifreleme algoritması geliştirebileceğini düşünebilir. Fakat, tasarım sürecinde verilen kritik güvenlik kararları vardır. Mesela, eğer üçüncü ve yedinci S-box'ların yerleri değiştirilirse, DES bir kat daha az güvenli olur.



Şekil III.I- I Feistel yapısı

**Soru:** DES'in çalışmasındaki ilk ve son permütasyonların güvenlik açısından bir değeri var mıdır?

## DES Çelişkisi

1970'leri başına kadar, kriptografi üzerinde askeri olmayan çalışmalar tesadüfiydi. DES geliştirilen ilk ticari şifrelerden biridir. DES önceden IBM'de bir ekip tarafından geliştirilen Lucifer şifresine dayanmaktadır. DES'in ABD devlet standardı olarak adaptasyonu kriptografiye olan kamusal bir ilgi doğurdu, bu sayede de günümüzde bilinen anlamıyla kriptografi doğmuş oldu. Diğer taraftan, DES şu iki sebepten dolayı pek çok çelişkiler doğurdu:

- Lucifer'in yapısı ciddi biçimde değiştirilmişti, ve DES'in tasarım sebebi asla kamuya açıklanmamıştı.
- Gizli anahtar boyutu 128-bitten 56-bite düşürülmüştü.

Anahtar aslında 64-bit gibi görünmekteydi, fakat 8 bitte her bir bit hata kontrolü amacıyla kullanılıyordu (tek eşitlik kontrolü). Bununla birlikte, anahtar boyuna olan eleştiriler DES'in pek çok uygulamada en sık kullanılan şifre olmasını engellemedi.

Tasarımından bu yana, DES'i geniş kapsamlı aramadan daha hızlı bir şekilde kırabilecek üç önemli saldırı keşfedildi: diferansiyel kriptanaliz, lineer kriptanaliz ve geliştirilmiş Davies saldırısı. Yine de, bu saldırılar analiz için  $2_{40}$ 'dan fazla metin gerektirdikleri için pek tehdit olarak algılanmadılar. İlginç bir durum ise DES'in tasarımcılarının diferansiyel kriptanalizden haberdar olmaları, ve bu saldırılara karşı koymak için DES'i tasarlamış olmalarıydı. DES'in tasarım kriterinin gizli tutulma sebebi de zaten buydu. Bu sırların pek çoğu diferansiyel kriptanalizin gelişimi ile kamuya açıldı, ve tasarımcılar tarafından da doğrulandı.

Şu anda DES'in kriptanalizine en tehlikeli yaklaşım hala geniş kapsamlı anahtar aramadır. 1977'de, özel olarak DES-şifresi çözen paralel bir bilgisayarın, 20 milyon \$'lık bir maliyet ile anahtarı tek günde bulabileceği öne sürüldü (fakat o zaman bu miktar sadece NSA benzeri organizasyonlar için "karşılabilir"di)." Günümüzde, DES'i standart bilgisayarlarla kırmak bile mümkündür. (değişikliği zamanı! :). DES zaman içerisinde ancak daha az güvenli olacaktır, çünkü anahtarı kısadır.

### Kısa Anahtar Sorununun Çaresi

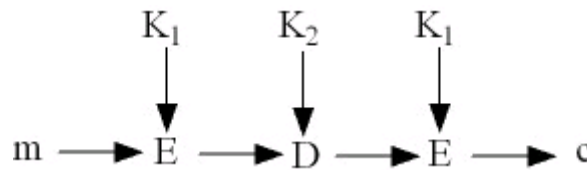
DES'in kısa anahtar problemi uzun zamandır bilinmekte. Bu problemin farklı bazı çözümleri var:

- Triple-DES (3DES)
- DESX
- Bağımsız alt anahtarlar
- Yavaş-anahtar planı
- Anahtara bağlı S-boxlar

3DES'i, yani en sık kullanılan çözümü burada açıklayacağız. Sağdaki figürde de görüldüğü üzere, 3DES iki farklı anahtar ile,  $K_1$  ve  $K_2$  üçlü şifreleme (Şifrele-Şifre Çöz-Şifrele) yapmaktadır.

Üç yerine iki anahtar kullanılmasının sebebi 112 bit anahtarların kaba kuvvet saldırılarına dayanıklılık konusunda yeterli görülmesidir.  $K_2$  nin şifre çözme modunda kullanılmasının sebebi  $K_1 = K_2$  durumunda sistemi DES'e eşit kılmaktır, bu sayede bu sistem DES sistemi ile birlikte çalışabilecektir.

**Soru:** Daha uzun anahtarlı yeni bir şifre kullanmanın DES'in etkin anahtar uzunluğunu arttırmaya göre dezavantajı ne olabilir?



Şekil III.I- II 3DES şifreleme sırası.

## Gelişmiş Şifreleme Standardı (Advanced Encryption Standard-AES)

3DES'in problemi hızı. 3DES çok yavaş. NIST'nin (National Institute of Standards and Technology) DES'in sonunun geldiğine ve yeni bir standartla değiştirilmesi gerektiğine karar verme sebebi de bu. NIST, 1997 yılında, yeni bir şifre tasarlamak veya tasarlatmak yerine, kriptografi dünyasından öneriler istedi. Bu yol ile, yeni standardın güvenliği ile alakalı spekülasyonlar da sona erdirilebilecekti.

Kriptografi dünyasında bir hayli uzun süren araştırmalar ve tartışmalardan sonra, NIST Rijndael olarak bilinen algoritmayı seçti, ve AES olarak standartlaştırdı (26 Kasım 2001).

### AES'in Tanımı

AES DES'ten farklı bir yapı kullanmaktadır. Bir Feistel yapısı değildir. AES ikame-permütasyon kullanan, 128-bit bloklu, 128/192/256 bit anahtar boyutlarına 10/12/14 döngü ile izin veren bir ağ şifresidir. AES'in şifreleme işlemleri aşağıda özetlenmiştir:

- İlk işlem 16-bytelık düzmetini 16-bytelık döngü anahtarı ile XORlamaktır.
- Daha sonra her 16 byte, 8-bit girdileri 8-bit çıktılara çeviren S-box tablolarına birer indeks olarak kullanılır.
- Bunu byte konumlarının permütasyonu izler.
- Son olarak, bytelar doğrusal bir karıştırma fonksiyonu ile dört gruba ayrılırlar.

AES'de, DES'teki fonksiyonel blokların bazılarını görebiliriz. (XORlar veriye anahtar bilgiler ekler, S-boxlar lineerliği engeller, ve byte karıştırma ve gruplara ayrılma difüzyon sağlar). AES şifrenin her bölümü için farklı bir görev bulunduran temiz bir tasarıma sahiptir. Şu ana kadar AES'de bir güvenlik açığı bulunamadı, fakat kimse ileride de bulunamayacağından emin olamaz.

**Soru:** Yukarıdaki paragrafta, "difüzyon" ve "lineer olmama" terimleri neyi ifade eder?

### Blok Şifre Modları

Blok şifreler sadece sabit-boyutlu blokları şifrelerler. Eğer bir blok boyutundan daha kısa bir düzmetini şifrelemek istiyorsanız, basitçe düzmetinin sonuna boşluklar ekleyebilirsiniz.

Blok boyutundan büyük metinler için, blok şifreleme modu kullanmalısınız. En basit ve malesef en az güvenli operasyon modu Elektronik Kod Kitabı (Electronic Code Book-ECB) olup, mesajları blok boyutlarına bölüp her bloğu gizli anahtar ile şifreleme dayanır. ECB kullanıldığında, eğer bir mesaj aynı metinlerden barındırıyorsa, bu metinlere denk düşen şifre metinler de aynı olacaktır. Bu problem, yandaki figürde de belirtildiği üzere, saldırganın önemli bilgiler sağlayabilir. ECB ile ilgili bir sorun da saldırganının blokların yerlerini kendi avantajına olacak şekilde düzenleyebiliyor olmasıdır.

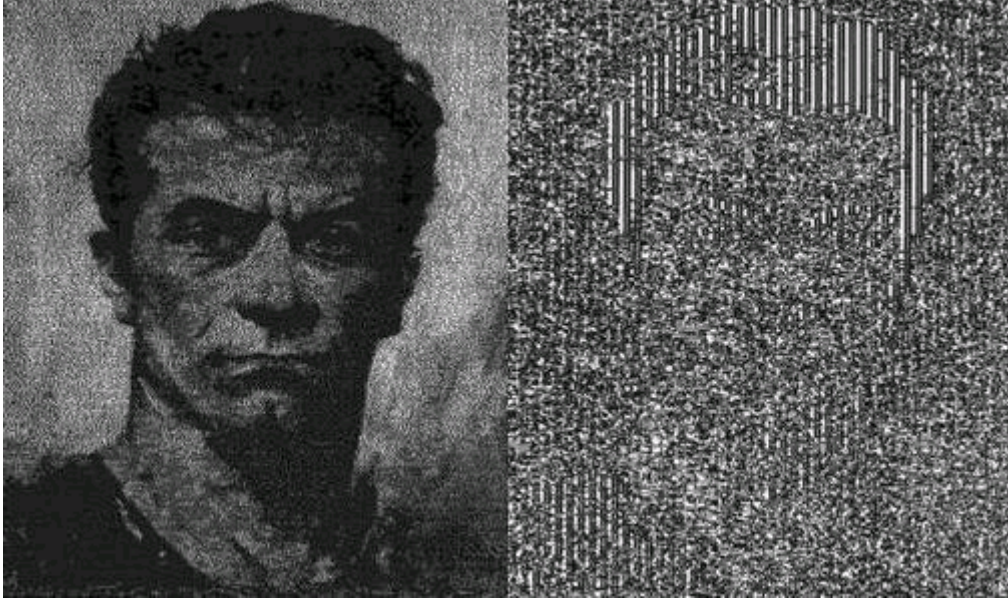
Başka operasyon modları ECB ile gelen problemleri çözmek için öne sürülmüştür:

- Şifre Blok Zincirleme (Cipher Block Chaining-CBC): CBC'de, ilk düzmetin bloğu rastgele bir sayı ile XORlanır (bu sayıya başlama vektörü (initialization vector) veya IV de denir). Figürden görülebileceği üzere, her takip eden düzmetin bloğu önceki

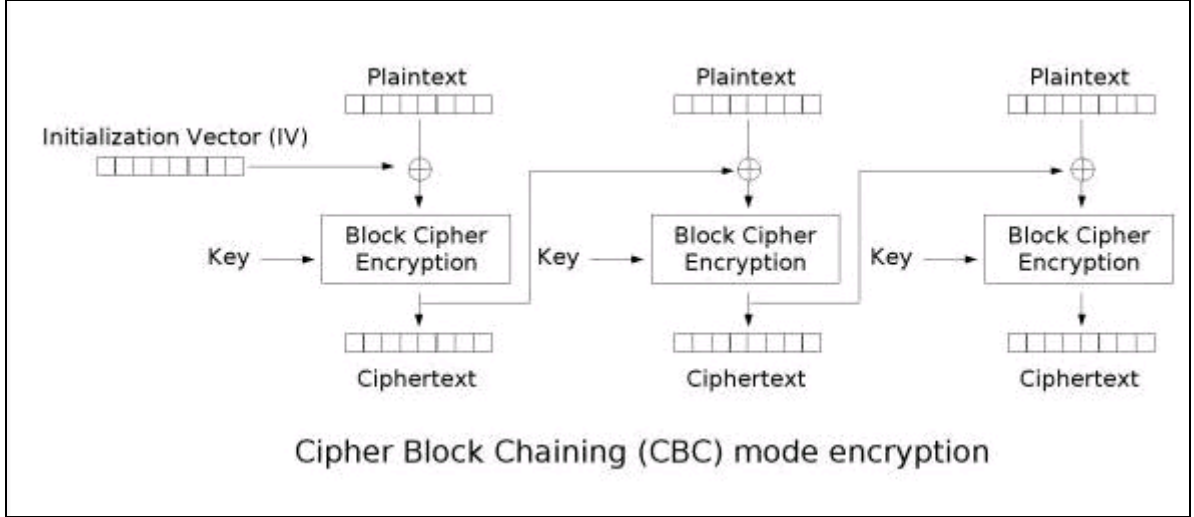
şifremetin bloğu ile XORlanır. Bu hareket sayesinde aynı düzmetin aynı şifremetine sahip olmayacaktır.

- Çıktı Geribesleme Modu (Output Feedback Mode-OFM): OFM'nin işlemesi akış şifrelerinininki gibidir.Şifreleme mesaj bloklarını OFM tarafından yaratılan tek seferlik pad blokları ile XORlama üzerine yapılır. İlk OTP bloğu bir başlama vektörünün şifrenmesi ile ortaya çıkarılır.Daha sonra, önceki OTP blokları şifrenerek sonraki OTPler elde edilir.
- Sayaç Modu (Counter Mode-CM): OFM'ye benzer, tek fark ikinci OTP bloğunun IV+1'in şifrenmesi, üçüncü OTP'nin IV+2'nin şifrenmesi vs. ile elde edilmesidir.
- Şifre Geribesleme Modu (Cipher Feedback Mode-CFB): OTP bloğu önceki şifremetin bloğunu şifreleyerek elde edilir.

Unutmayın ki, OFB ve CM modlarında, kriptografi önceden hesaplanabilir. Mesaj şifrelemeye hazır olduğunda, tek yapılması gereken XORlamaktır.



Şekil III.I- III Düzmetin-Şifremetin görünümler



Şekil III.I- IV CBC mod şifreleme.

**Soru:** CBC modununun çözemeyeceği güvenlik açıkları neler olabilir?

### Akış Şifreleri

Akış şifreleri blok şifrelerden şu bağlamda farklıdır:

- Şifreleme boyunca sürekli olarak değişen gizli bir “durum”a sahiptirler.
- Genellikle bloklar yerine bit akımları üretirler.

Bu yüzden akış şifrelerinin iki ana bölümü:

- Durum-değiştirme fonksiyonu
- Filtre (akış şifresinin çıktısını üretir)

Akış şifreleri genellikle blok şifrelerden daha verimli bir şekilde üretilebilirler. Bu yüzden, akış şifreleri şifreleme dünyasında uzun süre hüküm sürmüşlerdir. Hızlı blok şifrelerinin belirmesi ve blok şifrelerin CM, OFB veya CBC modlarında akış-benzeri davranabilmeleri ilginin blok şifrelere kaymasına sebep olmuştur. Üstelik, güvenlik güvenli bir akış şifresi tasarlamak güvenli bir blok şifre tasarlamaktan çok daha zordur.