T.C.

YOZGAT BOZOK ÜNİVERSİTESİ

MÜHENDİSLİK MİMARLIK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ

2020-2021 GÜZ – BAHAR YARIYILI

SİBER GÜVENLİĞE GİRİŞ DERSİ

Konu: Simetrik Şifreleme Kavramının Uygulamalı Anlatımı

ESRA YÜCE

16008117051

İçindekiler

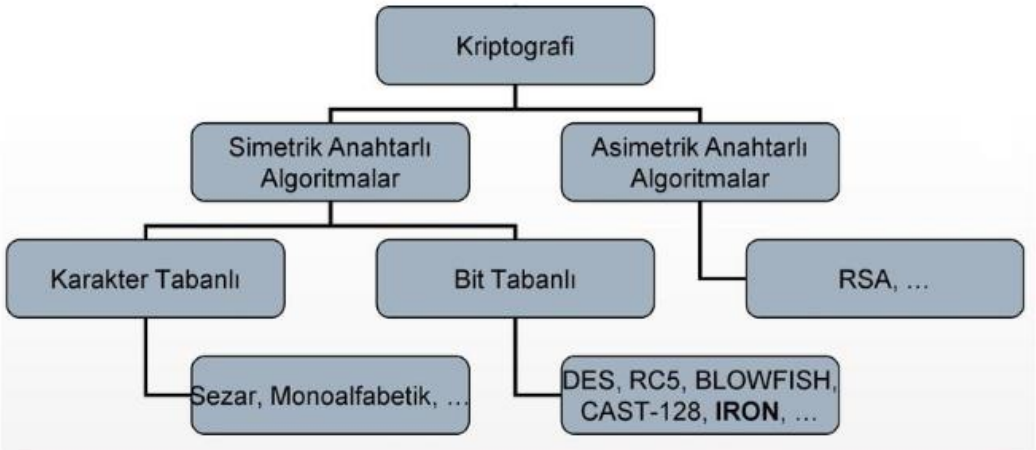
1. Giriş 3
2. Simetrik Şifreleme 3
2.1. Simetrik Şifreleme Algoritması Türleri 5
2.2. 3 Yaygın Simetrik Şifreleme Algoritması 6
2.2.1. DES 6
2.2.2. 3DES 10
2.2.3. AES 11
Simetrik Şifreleme Uygulaması 12
Kaynakça 17

1. Giriş

Kriptografi, en basit haliyle mesajları korumak için kodlar ve şifreler kullanma bilimidir. Şifreleme, mesajları yalnızca hedeflenen alıcının mesajın anlamını anlamasına izin vermek amacıyla kodlamaktır. Bu iki yönlü bir işlevdir (mesaja yaptığınız karıştırmayı geri alabilmeniz gerekir). Bu, geçiş halindeki verileri korumak için tasarlanmıştır.

Veri şifrelemenin ilk biçimleri ilkeldi; bazıları cümledeki harfleri değiştirmeyi içeriyordu. Bu, tüm cümleyi kesinlikle okunamaz hale getirdi ve dökülen karakterlerin ne anlama geldiğini anlamak için çok zaman gerekiyordu.

Ne yazık ki, analizdeki gelişmeler, özellikle otomatik analizler ve çok güçlü bilgisayarlar, bu tür şifrelerin kırılmasını çok kolaylaştırdı. Buna yanıt olarak çok güçlü, karmaşık algoritmalar geliştirildi. Bunlar iki temel şifreleme türüne ayrılabilir; Simetrik Şifreleme ve Asimetrik Şifrelemedir. Bu çalışmada Simetrik Şifreleme ele alınacaktır.



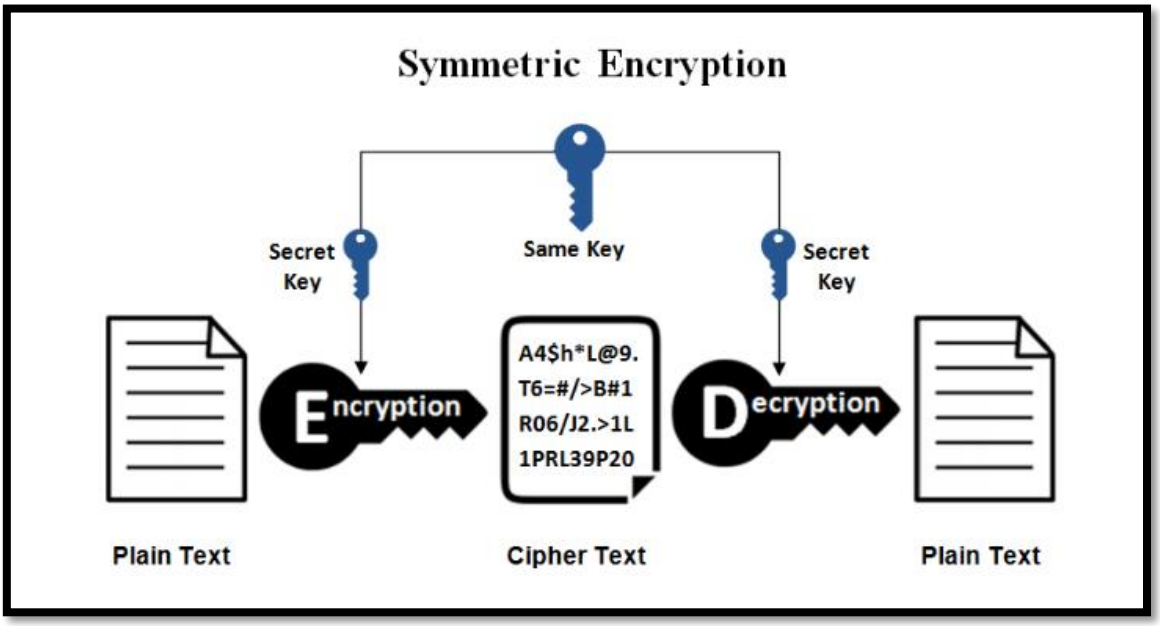
2. Simetrik Şifreleme

Simetrik şifreleme, mesajı şifrelemek ve şifresini çözmek için aynı anahtarın kullanıldığı bir şifreleme biçimidir. Bu, bir mesajı şifrelemek için bir anahtar ve mesajın şifresini çözmek için başka bir anahtar kullanan asimetrik veya açık anahtar şifrelemeden farklıdır.

Simetrik şifreleme, bir elektronik mesajın içeriğini gizlemek için belirli bir şifreleme anahtarı kullanan bir tür bilgisayarlı kriptografidir. Veri dönüşümü, özel bir anahtarla birlikte matematiksel bir prosedür kullanır ve bu, şifresini çözmek için doğru araçlara sahip olmayan biri için bir mesajı anlamlandırma potansiyel başarısızlığına neden olur. Simetrik şifreleme iki yönlü bir algoritmadır çünkü mesajın şifresi çözülürken matematiksel prosedür aynı özel anahtarın kullanılmasıyla birlikte geri çevrilir [1].

Örneğin Bob'un yabancı bir ülkede gizli bir görevde bulunan gizli bir casus ajan olduğunu varsayalım. Alice ise, onu izleyen ve ona rehberlik eden dava görevlisidir. Etrafı düşmanlarla çevrili olan Bob, Alice'e gönderebilmek için bilgi topluyor. Ancak büyük bir endişesi var: Alice'e gönderdiği veriler düşmanlar tarafından yakalanabilir ve ifşa edilebilir.

Bunun olmasını önlemek için Alice, Bob'a gizli bir anahtar verir ve göndermeden önce tüm bilgileri şifrelemesini ister. Bob kabul eder ve verileri şifrelemek için bu anahtarı kullanır. Alice aynı anahtara sahiptir ve gizli bilgileri görüntülemek için verilerin şifresini çözmek için aynı anahtarı kullanır. Bu şekilde Bob'un kimliği bir sır olarak kalır ve veriler Alice'e aktarılır [2].



Simetrik Şifreleme

Bu, bilgileri şifrelemek ve deşifre etmek için yalnızca bir özel anahtardan oluşan en basit şifreleme biçimidir. Simetrik şifreleme eski ve iyi bilinen bir uygulamadır. Bir sayı, kelime veya rastgele harflerden oluşan bir dizi olabilen özel bir anahtar kullanır. İçeriği belirli bir şekilde değiştirmek için bir mesajın düz metni ile karıştırılır. Gönderen ve alıcı, tüm mesajları şifrelemek ve deşifre etmek için kullanılan özel anahtarı bilmelidir.

Simetrik şifrelemenin avantajları şunlardır;

- Simetrik şifreleme oldukça hızlıdır.
- Donanımla beraber kullanılabilirler.
- Anahtarları oldukça kısadır.
- Daha güçlü şifrelerin oluşturulmasında aracı olarak kullanılabilirler.
- Kullanımları gayet anlaşılır ve kolaydır.

Simetrik şifrelemenin dezavantajları şunlardır;

- Anahtarın saklanması ve taraflara ulaştırılması zordur, güvenlik riski içerir.
- Büyük ağlarda çok fazla anahtara ihtiyaç duyulur, ölçeklenebilir değildir. n kullanıcılı bir sistemde [n*(n-1)/2] anahtar vardır.
- Kimlik doğrulama işlevi görülmez. Anahtara sahip olan herkes olaya dahil olur.
- Bütünlük sağlaması yoktur. Anahtarı ele geçiren kişi veriyi değiştirmiş olabilir.
- İnkar edilememezlik sağlamaz.

Kriptografide kullanılan bazı temel kavramlar;

Plaintext: Orijinal, düz metin.

Ciphertext: Şifrelenmiş metin.

Cipher: Düz metni, şifrelenmiş metne çeviren algoritma. Şifreleme algoritması.

Encipher (encrypt): Düz metni şifrelenmiş metne çevirme.

Decipher (decrypt): Şifreli metinden düz metni kurtarma.

2.1. Simetrik Şifreleme Algoritması Türleri

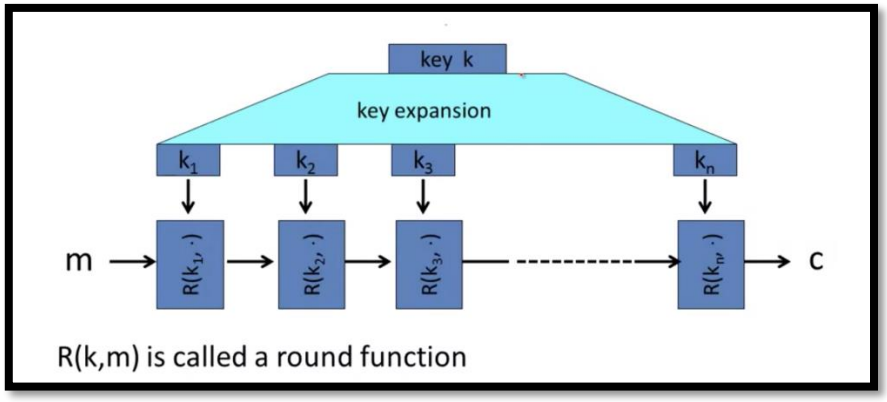
Bir web sitesine güvenli bir şekilde bağlanıldığında gerçekleşen simetrik şifreleme sırasında, bunun gerçekleşmesi için simetrik bir şifre kullanılır. Simetrik şifrelerin iki alt kategorisi vardır: Blok Şifreleme Algoritmaları ve Akış Şifreleme Algoritmaları [4].

Blok Şifreleme

Bu tür şifrelemede, düz metin verileri bloklar olarak bilinen sabit uzunlukta bit gruplarına ayrılır (bunlar tipik olarak zincirleme olarak bilinen bir işlemle bağlanır). Her blok daha sonra bir birim olarak şifrelenir ve bu da bu işlemi biraz yavaşlatır. Ve bir bloğu tamamen doldurmak için yeterli veri yoksa, blokların sabit uzunluk gereksinimlerini karşıladığından emin olmak için " padding " kullanılır.

İdeal blok şifreleme, pratik olmayan devasa bir anahtar uzunluğuna sahiptir, bu nedenle birçok modern şifrenin, onları kullanılabilir hale getirmek için anahtar boyutlarını küçültmesi gerekir. Ancak Asimetrik şifrelemenin aksine, simetrik şifreleme anahtarı boyutları veri bloklarının boyutunu belirlemez.

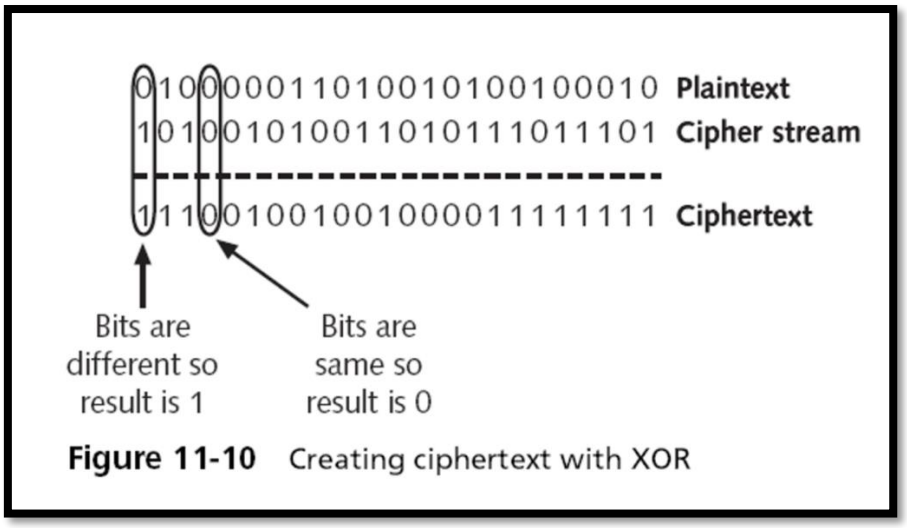
Modern simetrik şifreleme algoritmalarının çoğu, blok şifreleme türüne girer ve bu tür şifrelerin daha geniş kullanım ve uygulama fırsatları vardır.



Blok Şifreleme

Akış Şifreleme

Bu tür şifreleme ile düz metin verilerini her seferinde bir bit şifreler. Bu nedenle veriler, blok şifrelemede olduğu gibi yığınlar yerine bir akışta işlenir. Bu, sürecin daha az yoğun ve daha hızlı gerçekleştirilmesini sağlar.



Akış Şifreleme

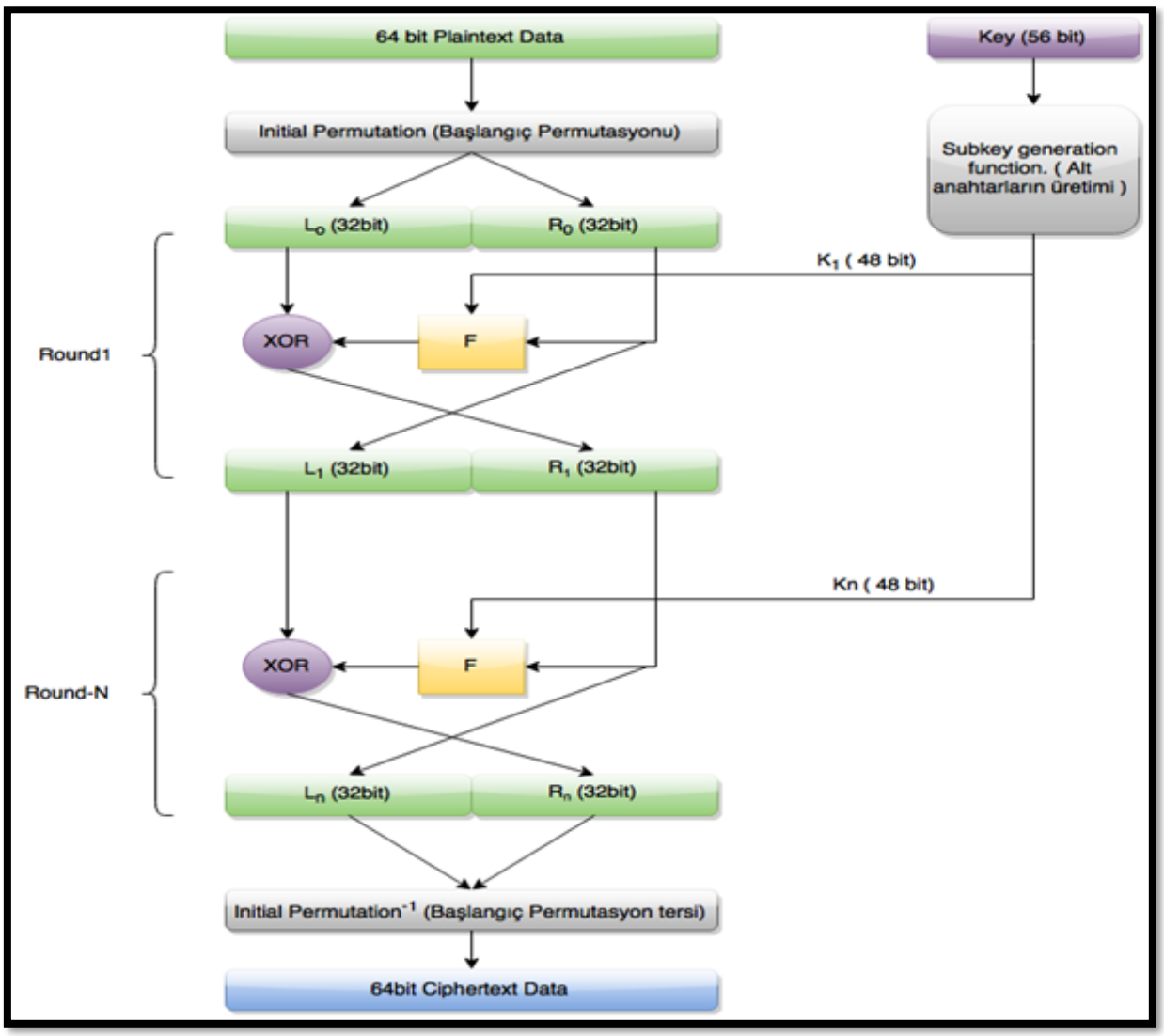
2.2. 3 Yaygın Simetrik Şifreleme Algoritması

2.2.1. DES

1976'da tanıtılan DES (veri şifreleme standardı), en eski simetrik şifreleme yöntemlerinden biridir. IBM tarafından hassas, sınıflandırılmamış elektronik devlet verilerini korumak için geliştirildi ve resmi olarak 1977'de federal kurumlar tarafından kullanılmak üzere kabul edildi. DES, 56 bitlik bir şifreleme anahtarı kullanır ve bu, Horst Feistel adlı bir kriptograf

tarafından tasarlanan Feistel Yapısını temel alır. DES şifreleme algoritması, TLS (taşıma katmanı güvenliği) sürüm 1.0 ve 1.1'de bulunanlar arasındaydı.

DES'in çalışması aşağıdaki tablodan adım adım incelenirse;

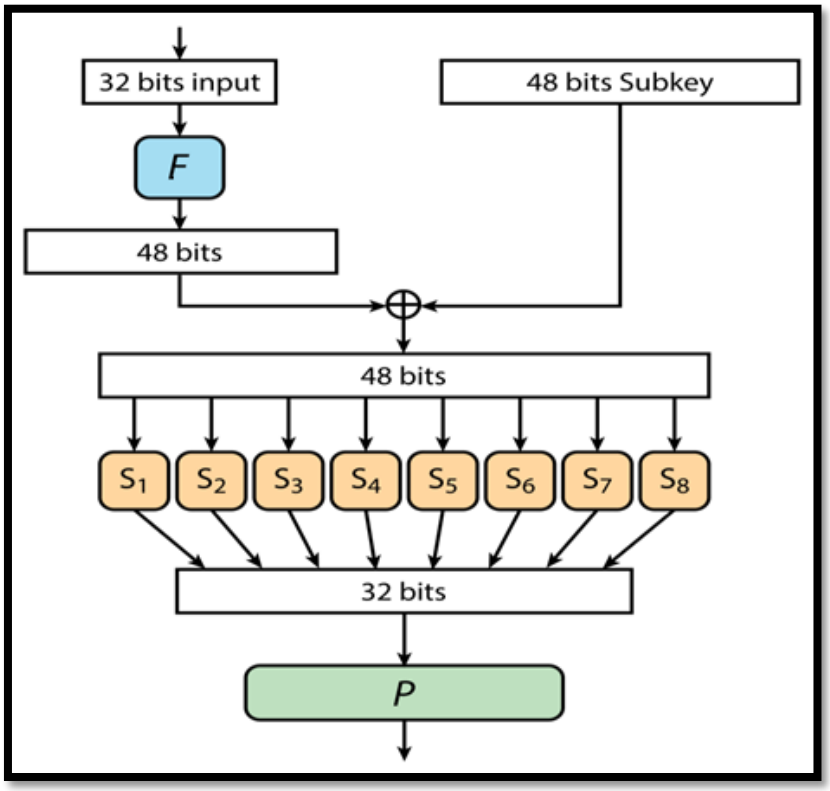


DES algoritması akış şeması

Şifrelenecek metin 64 bit uzunluğunda bloklara bölünür ve ilk olarak başlangıç permutasyonu tablosundan geçirilir. Sol ve sağ olarak 2 parçaya ayrılır. Daha sonra bu parçalar ayrı ayrı bir takım işlemlerden geçer. Bu işlemler sağ parça ve anahtar F fonksiyonuna girer ve sonrasında diğer parça ile XOR işlemine uğrar. Son olarak parçaların yerleri değiştirilir.

DES te bu uygulama standart olarak 16 defa tekrar edilir.

DES Fonksiyonu

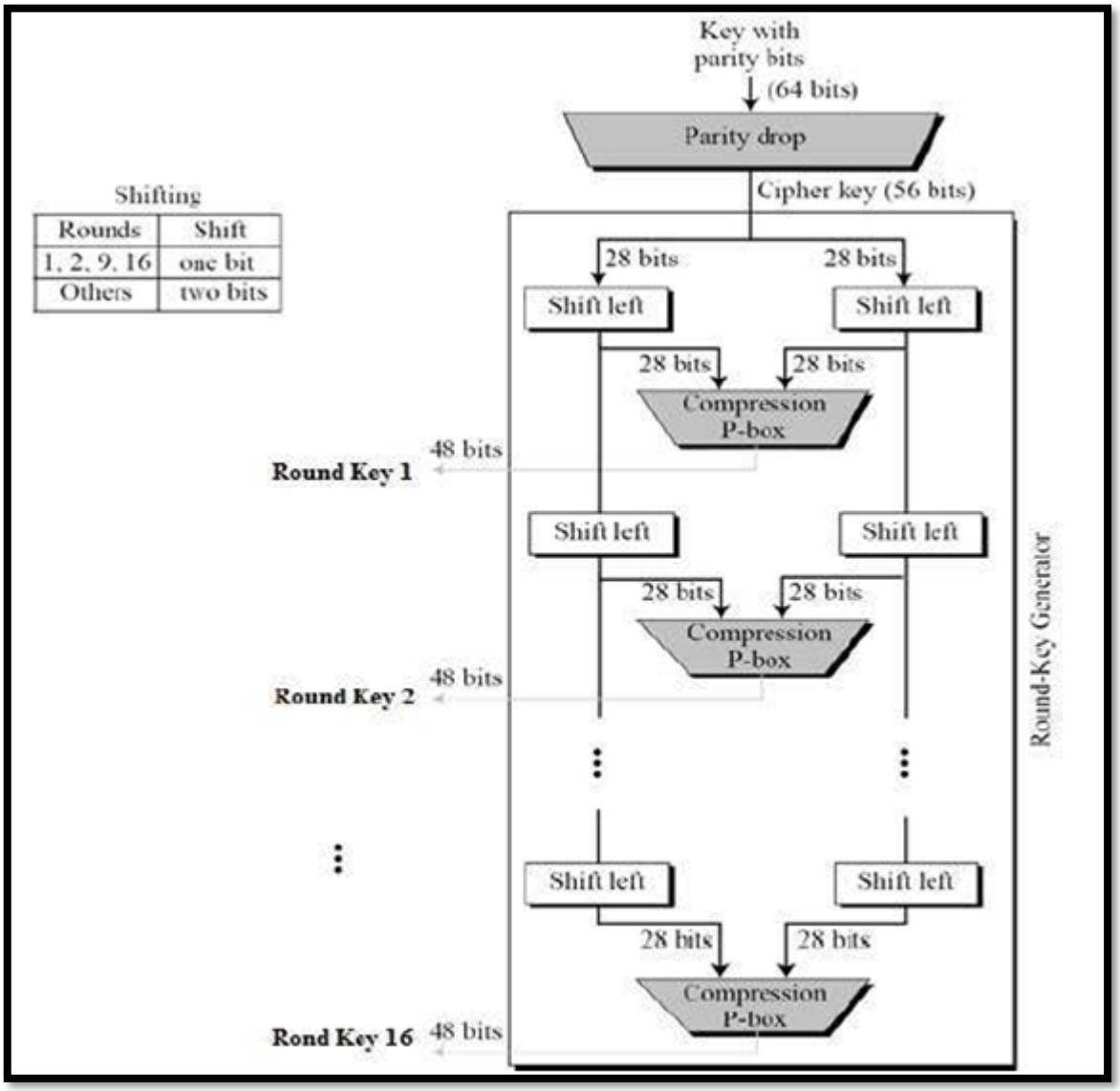


DES fonksiyonu

Buradaki F fonksiyonunun içerisinde yaptığı işlemler şu şekildedir;

Genel metnin ikiye bölünmüş parçasından birini ve 48 bitlik anahtarı girdi olarak alır. İlk olarak 32 bitlik veriyi genişletme (expansion) işlemine tabi tutar yani bit birden fazla yerde geçerek boyut artırılmıştır. Burada oluşan genişletilmiş metin ile anahtar XOR yapar ve 48 bitlik bir sonuç elde edilir. Bu sonuç metni altışar bitlik 8 ayrı bloğa bölünür. Bu bloklardaki bitler dörder bite indirilir bunlara S-kutuları (s-box) denilmektedir. Yani DES her biri 6 bit giriş ve 4 bit çıkış şeklinde olan 8 S kutusu kullanır. Son olarak 32 bitlik bir veri elde edilir.

Anahtar Üretimi



DES için anahtarlama

İlk olarak 64 bit boyutunda alınan anahtarın 8 biti parity biti olarak ayrılır.

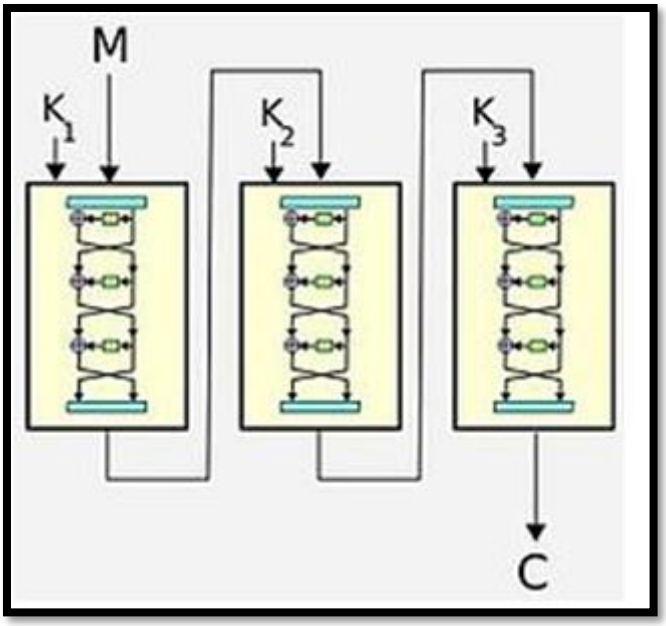
Anahtar üreteci, 56 bitlik şifreleme anahtarından 48 bit anahtar oluşturur. Her bir adım 28 bite bölünmüş ve her seferinde sola kaydırma yapılarak anahtarın değişmesi sağlanmıştır. Her bir geçiş için bir anahtar üretildiğinden yukarıdaki tabloda 16 adım vardır.

Günümüzde, birçok güvenlik araştırmacısı tarafından kırıldığı için DES artık kullanılmamaktadır. 2005 yılında DES resmi olarak kullanımdan kaldırıldı ve yerini, birazdan bahsedeceğimiz AES şifreleme algoritması aldı. DES'in en büyük dezavantajı, düşük şifreleme

anahtarı uzunluğuydu ve bu da kaba zorlamayı ona karşı kolaylaştırdı. Günümüzde en yaygın kullanılan TLS protokolü olan TLS 1.2, DES şifreleme yöntemini kullanmaz [5].

2.2.2. 3DES

Adından da anlaşılacağı gibi, 3DES (aynı zamanda üçlü veri şifreleme algoritması anlamına gelen TDEA olarak da bilinir), piyasaya sürülen DES algoritmasının yükseltilmiş bir sürümüdür. 3DES, DES algoritmasının dezavantajlarının üstesinden gelmek için geliştirilmiş ve 1990'ların sonlarından itibaren kullanıma girmiştir. Bunu yapmak için DES algoritmasını her veri bloğuna üç kez uygular. Sonuç olarak, bu süreç 3DES'in kırılmasını önceki DES'e göre çok daha zor hale getirdi. Ayrıca finans sektöründe ödeme sistemleri, standartları ve teknolojisinde yaygın olarak kullanılan bir şifreleme algoritması haline geldi.



3DES algoritması

İlk DES bloğundan çıkan şifreli metin ikinci DES bloğunun şifrelenecek olan metini olur. Aynı şekilde üçüncü DES bloğunun şifrelecek metni ikinci DES bloğundan çıkan şifrelenmiş metindir. Her DES bloğu için aynı anahtar kullanılabileceği gibi üç ayrı anahtar da kullanılabilir.

TLS, SSH, IPsec ve OpenVPN gibi kriptografik protokollerin bir parçası haline geldi.

Tüm şifreleme algoritmaları nihayetinde zamanın gücüne yenik düştü ve 3DES de farklı değildi. 3DES algoritması içinde Sweet32 açığı araştırmacılar Karthikeyan Bhargavan ve Gaëtan Leurent tarafından keşfedildi. Bu keşif, güvenlik endüstrisinin algoritmanın

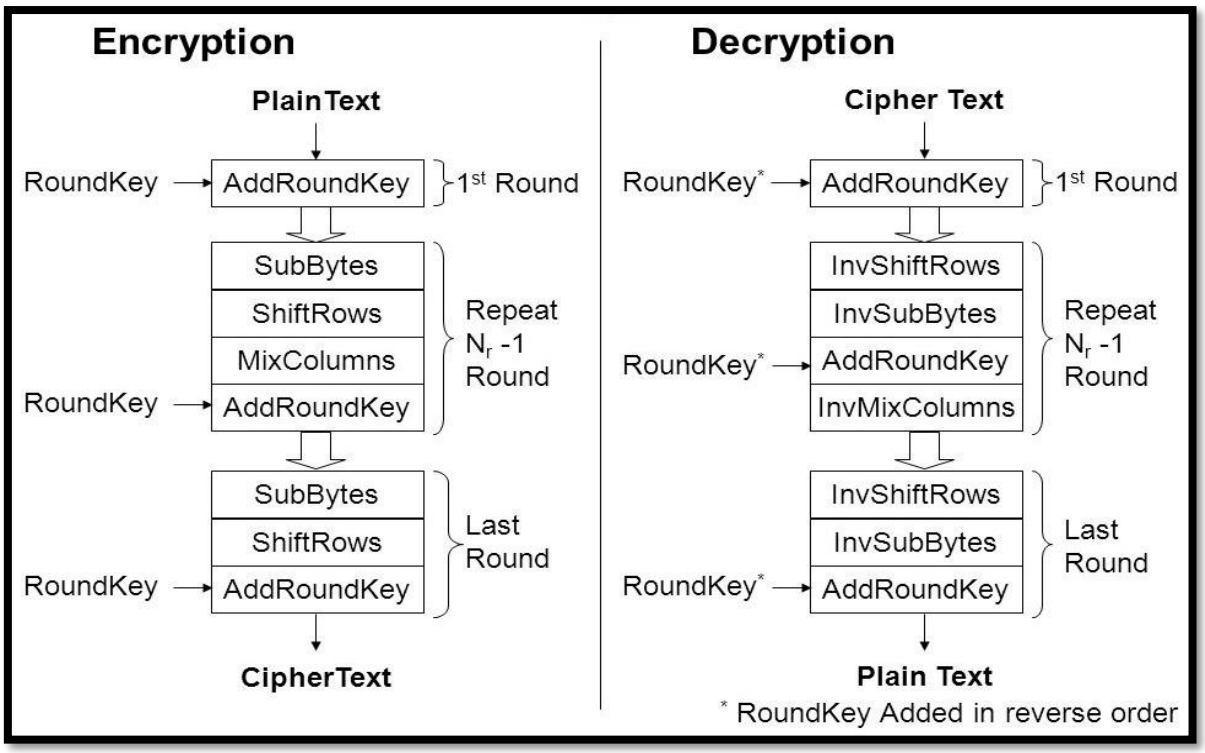
kullanımdan kaldırıldığını düşünmesine neden oldu ve Ulusal Standartlar ve Teknoloji Enstitüsü(NIST) , 2019'da yayınlanan bir taslak kılavuzda kullanımdan kaldırıldığını duyurdu.

Bu taslağa göre, 2023'ten sonra tüm yeni uygulamalarda 3DES kullanımı durdurulacak. SSL / TLS protokolleri için en son standart olan TLS 1.3'ün de 3DES kullanımını durdurduğunu belirtmekte fayda var [5].

2.2.3. AES

"Gelişmiş Şifreleme Standardı" anlamına gelen AES, en yaygın kullanılan şifreleme algoritmaları türlerinden biridir ve DES algoritmasına alternatif olarak geliştirilmiştir. Rijndael olarak da bilinen AES, 2001 yılında NIST tarafından onaylandıktan sonra bir şifreleme standardı haline geldi. DES'ten farklı olarak AES, farklı anahtar uzunluklarına ve blok boyutlarına sahip şifrelerden oluşan bir blok şifreleme ailesidir.

AES, ikame ve permütasyon yöntemleri üzerinde çalışır. Önce düz metin verileri bloklara dönüştürülür ve ardından şifreleme anahtarı kullanılarak şifreleme uygulanır. AES 128 bitlik blok boyutlarında çalışır ve üç anahtar uzunluğu seçeneği vardır: 128, 192 veya 256 bit. Şifreleme işlemi, alt baytlar, satır kaydırma, sütunları karıştırma ve yuvarlak anahtar ekleme gibi çeşitli alt işlemlerden oluşur. Anahtarın boyutuna bağlı olarak, bu tür 10, 12 veya 14 tur gerçekleştirilir. Son turun, verileri şifrelemek için gerçekleştirilen diğer tüm alt işlemler arasında sütunları karıştırma alt sürecini içermediğini belirtmek gerekir.



AES Şifreleme Algoritmasını Kullanmanın Avantajı

Tüm bunların özü, AES'in güvenli, hızlı ve esnek olduğunu söylemektir. AES, DES'e kıyasla çok daha hızlı bir algoritmadır. Birden fazla anahtar uzunluğu seçeneği, sahip olduğunuz en büyük avantajdır; tuşlar ne kadar uzunsa, onları kırmak o kadar zor olur.

Günümüzde, AES en yaygın kullanılan şifreleme algoritmasıdır - aşağıdakiler dahil birçok uygulamada kullanılmaktadır:

- Kablosuz güvenlik,
- İşlemci güvenliği ve dosya şifreleme,
- SSL / TLS protokolü (web sitesi güvenliği),
- Wi-Fi güvenliği,
- Mobil uygulama şifreleme,
- VPN (sanal özel ağ) vb.

Ulusal Güvenlik Ajansı (NSA) dahil birçok devlet kurumu, hassas bilgilerini korumak için AES şifreleme algoritmasına güveniyor [5].

Simetrik Şifreleme Uygulaması

Uygulama için Google Colaboratory ortamı kullanıldı. Google Colaboratory, makine öğrenimi araştırmalarına ve çalışmalarına yardımcı olmak için oluşturulan bir projedir. Kurulum gerektirmeyen ve tamamen bulutta çalışan bir Jupyter notebook ortamıdır.

Colab ile popüler Python kitaplıklarının tüm avantajlarından yararlanarak veriler analiz edilip görselleştirilebilir.

Uygulamaya ait kodlar ve açıklamaları aşağıda verilmiştir.

Adım 1:

```
!pip install pycrypto
```

Python modülü colab ortamına dâhil edildi.

Adım 2:

```
from Crypto.Cipher import DES,AES
from Crypto import Random
```

Şifreleme işlemi için Crypto Kütüphanesi altındaki Cipher'ın içerisinden DES ve AES algoritmaları import edildi.

Crypto kütüphanesinden Random modülü import edildi.

Adım 3:

```
desKey = b"SECURITY" # DES 64 bit
aesKey = b"SECURITYSECURITY"  # AES 128 bit
```

DES ve AES algoritmaları için anahtar uzunlukları ayrı ayrı belirlendi.

DES için 64 bit ve AES için 128 bit uzunluğunda anahtar girildi.

Adım 4:

```
metin = input("Şifrelenecek Metni Giriniz: ")
plain = bytes(metin,encoding = "utf-8")
```

Kullanıcıdan şifrelenmesi istenen metnin girilmesi istendi.

Girilen metin bytes türüne dönüştürülerek plain değişkenine atandı.

Çıktı:

```
Şifrelenecek Metni Giriniz: Siber Güvenliğe Giriş Dersi
```

DES Şifreleme Algoritması:

Adım 5:

```
kalan = len(plain) % 8
if kalan != 0:
  for i in range(8-kalan):
    plain += b"-"
else:
  print("Metin Tam Şifrelenebilir")
```

DES algoritması metnin uzunluğunu 8 ve 8'in katları bytes şeklinde kabul etmektedir. Bu durumda metin boyutu eksik ya da fazla ise metin boyutu 8'in katlarına tamamlanır. Tamamlama işlemi eksik bitler için '-' koyacak şekilde ayarlandı.

Adım 6:

```
iv = Random.new().read(DES.block_size)
desCipher = DES.new(desKey, DES.MODE_ECB,iv)
desCipherMsg = desCipher.encrypt(plain)
```

DES için şifreleme işlemi başlatıldı ve işlem gerçekleşti.

Adım 7:

```
msgDes = desCipher.decrypt(desCipherMsg)
```

DES ile şifrelenmiş metnin şifresi çözüldü.

Adım 8:

```
sifreliMetin1 = desCipherMsg.decode('latin-1')
sifreliMetin1
```

Şifreli metin ekrana yazdırıldı.

Çıktı:

```
'ÍÇ!êß^IhjzJç¾W|opðÕÏô=ÇlÏ¯.å\x8aÐûO'
```

Adım 9:

```
duzMetin1 = msgDes.decode()
duzMetin1 = duzMetin1.replace("-","")
duzMetin1
```

Metin boyutunu 8 ve 8'in katlarına tamamlamak için girilen '-' işareti şifresi çözülmüş metinden çıkartıldı.

Şifresi çözülen metin ekrana yazdırıldı.

Çıktı:

```
'Siber Güvenliğe Giriş Dersi'
```

AES Şifreleme Algoritması:

Adım 10:

```
kalan = len(plain) % 16
if kalan != 0:
  for i in range(16-kalan):
    plain += b"-"
else:
  print("Metin Tam Şifrelenebilir")
```

AES algoritması metnin uzunluğunu 16 ve 16'nın katları bytes şeklinde kabul etmektedir. Bu durumda metin boyutu eksik ya da fazla ise metin boyutu 16'nın katlarına tamamlanır. Tamamlama işlemi eksik bitler için '-' koyacak şekilde ayarlandı.

Adım 11:

```
iv = Random.new().read(AES.block_size)
aesCipher = AES.new(aesKey, AES.MODE_ECB,iv)
aesCipherMsg = aesCipher.encrypt(plain)
```

AES için şifreleme işlemi başlatıldı ve işlem gerçekleşti.

Adım 12:

```
msgAes = aesCipher.decrypt(aesCipherMsg)
```

AES ile şifrelenmiş metnin şifresi çözüldü.

Adım 13:

```
sifreliMetin2 = aesCipherMsg.decode('latin-1')
sifreliMetin2
```

Şifreli metin ekrana yazdırıldı.

Çıktı:

```
'\x8eTX>ð\x1a\x85Gxâ\xa01¤\rÍ¢0¡\x18_v\x0b)80Ç¿\x14Ä)K\x96'
```

Adım 14:

```
duzMetin2 = msgAes.decode()
duzMetin2 = duzMetin2.replace("-","")
duzMetin2
```

Metin boyutunu 16 ve 16'nın katlarına tamamlamak için girilen '-' işareti şifresi çözülmüş metinden çıkartıldı.

Şifresi çözülen metin ekrana yazdırıldı.

Çıktı:

```
'Siber Güvenliğe Giriş Dersi'
```

Kaynakça

1. https://medium.com/@hicranozkan/simetrik-ve-asimetrik-anahtarlı-şifreleme-algoritmaları-a60a4e0eb079

2. https://sectigostore.com/blog/types-of-encryption-what-to-know-about-symmetric-vs-asymmetric-encryption/

3. https://www.cryptomathic.com/news-events/blog/symmetric-key-encryption-why-where-and-how-its-used-in-banking#:~:text=Symmetric%20encryption%20is%20a%20type,used%20in%20the%20decryption%20process.

4. https://www.cryptomathic.com/news-events/blog/symmetric-encryption-algorithms-their-strengths-and-weaknesses-and-the-need-for-crypto-agility

5. https://kerteriz.net/modern-sifreleme-yontemleri-simetrik-asimetrik-sifreleme/