Karadeniz Teknik Üniversitesi

Bilgi Güvenliği ve Kriptoloji

Final Projesi

385961 Erdem Akgün

385960 Batırhan Berk Fil

385954 Mert Akın

Ocak 2022

Amaç

İnsanlar, tarih boyunca yalnızca istedikleri kişinin okumasını umdukları mesajları şifreleme yöntemini kullanarak göndermişlerdir. Günümüzde ise bu şifreleme işlemini bizim yerimize yapabilen bilgisayarlara sahibiz. Bugün gizli mesajlaşmanın da ötesine giden dijital şifreleme teknolojisi, mesajların yazarını doğrulamak gibi özel amaçlar için de kullanılabiliyor.

Bir kripto sistem;

şifreleme algoritması, anahtar, açık metin ve şifreli metinden oluşmaktadır. Günümüzde kullanılan modern şifreleme algoritmaları üç ana kategoriye ayrılmaktadır. Bunlardan ilki simetrik şifreleme algoritmalarıdır. Blok şifreleme algoritmaları bu kategoriye girer. Bu tür algoritmalarda şifreleme ve deşifreleme işlemeleri aynı anahtarı kullanır. Kullanılan anahtara gizli anahtar denir. İkinci karma şifreleme algoritmaları üçüncüsü ise asimetrik şifreleme algoritmalarıdır ve şifreleme için gizli anahtarı kullanırken deşifreleme için açık anahtarı, yani herkesin erişebileceği anahtarı, kullanır.

Proje İçeriği

1-) RSA Şifreleme Yöntemi

2-) DES Şifreleme Yöntemi

3-) Dijital İmza

1-) RSA Şifreleme Yöntemi

RSA algoritması asimetrik bir kriptoloji algoritmasıdır. Asimetrik çalıştığı için iki farklı anahtar kullanır. Bu anahtarlara Public Key ve Private Key denilir.

RSA mantığı, bir tam sayının çarpanlarına ayrılmasının onu yeni sayılarla çarpmaktan daha zor olduğu gerçeğine dayanmaktadır. Yeterince büyük ve birbirinden farklı olan iki asal sayının çarpımından oluşan bir base değer elde edilir. Ve diğer anahtar parametreleri de aynı iki asal sayıdan türetilir.

RSA algoritması için aşağıdaki adımlar uygulanır.

1-) Öncelikle p ve q değerleri belirlenir. Bu değerler asal sayı olmak zorundadır.

Değerler şifrenin güvenliği açısından olabildiğince yüksek verilmelidir.

2-) N değeri bulunur. (N=p*q), T(n) değeri bulunur. T(n)=(p-1)*(q-1)

3-) E değeri bulunur. E değeri 1<e<T(n) aralığında olmalıdır. T(n) ile aralarında asal olan herhangi bir sayı seçilebilir.

4-) D değeri bulunur. D değerini bulmak için d*e=1+0mod(T(n)) formülü kullanılır. Bu işlemde bulacağımız d sayısının modu 0 çıkmalıdır. Modu 0 bulunana kadar bu işlem devam eder.

5-) Public Anahtarımız = (N,E) , Private Anahtarımız = (N,D) olarak bulunur.

6-) Şifrelenecek olan metin girilir. Metin ASCII haline dönüştürülür.

7-) ASCII'ye çevirdiğimiz metnimizi şifrelemek için public anahtar kullanılır.

Formülümüz sifreliMetin=(metin^e)%n ile bulunur ve sifreliMetin ASCII halinden string haline dönüştürülür. Şifreleme tamamlanır.

8-) Şifrelenen metni tekrardan deşifre etmek için private anahtar kullanılır. Formülümüz metin=(sifreliMetin^d)%n ile bulunur. Bulduğumuz değer ASCII halinden string haline dönüştürülür ve metin tekrardan elde edilir.

```
p Sayısını Girin : 151
q Sayısını Girin : 47
N Sayısı :  7097
T Sayısı :  6900
E Sayısı :  4141
D Sayısı :  2761
Public Anahtar = ( 7097 4141 )
Private Anahtar = ( 7097 2761 )
Şifrelicek Metini Giriniz .nasılsın
Şifrelenicek Metnin ASCII Hali = [110, 97, 115, 305, 108, 115, 305, 110]
Şifrelenen Metnin ASCII Hali =  [580, 1225, 5426, 963, 4949, 5426, 963, 580]
Şifrelenmiş Metin =  ['ʁ', 'Ɦ', 'ג', 'σ', 'ד', 'ג', 'σ', 'ʁ']
Deşifrelenmiş Metnin ASCII Hali =  [110, 97, 115, 305, 108, 115, 305, 110]
Deşifrelenmiş Metin =  nasılsın
```

2-) DES Şifreleme Yöntemi

Açılımı Data Encryption Standart olan simetrik şifreleme algoritmasıdır.

DES, veri şifrelemek (encryption) ve şifrelenmiş verileri açmak (decryption) için geliştirilmiş bir standarttır.

DES yapısı itibari ile blok şifreleme örneğidir.

Yani basitçe şifrelenecek olan açık metni (plain text) parçalara bölerek (blok) her parçayı birbirinden bağımsız olarak şifreler ve şifrenmiş metni (cipher text) açmak içinden aynı işlemi bloklar üzerinde yapar.

DES algoritması için aşağıdaki adımlar uygulanır.

ANAHTAR ÜRETİMİ

1-) RSA'da paylaşılan anahtarımız hexadecimale çevrilir. Bu anahtarın uzunluğu 64 bit olmak zorundadır.

2-) Belirlenen anahtar Hexadecimal sayı sisteminden Binary sayı sistemine dönüştürülür.

3-) Binary sayı sistemine dönüştürülen 64 bitlik anahtar 8 biti parity bit olmak üzere parity bit tablosuna göre permute edilerek 56 bite indirgenir. (key_parity)

4-) 56 bite düşürülen anahtarımız 28 bitlik 2 parçaya ayrılır.(Sol ve Sağ)

5-) Anahtarın 28 bitlik 2 parçası her adımda sola kaydırma tablosundaki değerlere göre dairesel bir şekilde sola kaydırılır. (1,2,9 ve 16. adımlarda 1 bit, kalan adımlarda 2 bit kaydırılır.) Bu işlem 16 kez gerçekleşir

6-) Sola kaydırma işleminde her adımdan sonra 28 bitlik iki parça birleştirilir ve 56 bitlik 16 adet parça oluşturulur.

7-) Oluşturulan 16 adet 56 bitlik parça, anahtar sıkıştırma tablosuna göre permute edilerek 16 adet 48 bitlik adım anahtarına dönüştürülür ve algoritmanın anahtar üretim aşaması tamamlanır.

ŞİFRELEME

8-) 64 bit formatında metin alınır.

9-) Alınan metin Hexadecimal sayı sisteminden Binary sayı sistemine dönüştürülür.

10-) Dönüştürülen 64 bitlik sayı dizisi başlangıç permütasyonuna göre permute edilip sol ve sağ olmak üzere 2 parçaya ayrılır.

11-) 32 bitlik sağ parça(R0) genişletme tablosuna göre permute edilip 48 bite genişletilir.

12-) 48 bite genişletilen sağ mesaj ile 16 adet 48 bitlik anahtarlardan aynı aşama sırasına sahip anahtar(K1) XOR edilir.

13-) XOR işlemi sonrası elde edilen 48 bitlik veri 8 parçaya bölünür.

14-) 6 bitlik her parça ilgili sbox tan işleme tabi tutularak her sbox işlemi sonucunda decimal bir değer elde edilir

15-) Elde edilen decimal değerler binary sayı sistemine dönüştürülür ve 32 bitlik binary bir parça elde edilir.

16-) 32 bitlik verimiz permutasyon tablosuna göre permute edilir.(per)

17-) Elde edilen parça önceki sol parça ile XOR işlemine girerek çıkan sonuç sol parçanın değeri olur.

18-) Sol parça ve sağ parça yer değiştirir.

19-) Bu işlemler 16 defa tekrarlanır.

20-) Sol mesaj(L16) ile sağ mesaj(R16) birleştirilir.

21-) 64 bitlik bulduğumuz değer final permutasyon tablosuna göre permute edilir.(final_perm)

22-) Elde edilen sonuç hexadecimale çevrilip şifreleme işlemi tamamlanır.

```
Şifrele
İlk Permütasyon işleminden sonra : CC00CCFFF0AAF0AA
Adım  1   F0AAF0AA   6780CF9C   C3BF4A01A066
Adım  2   6780CF9C   C70D46B8   F222FF504399
Adım  3   C70D46B8   CB365471   BDD634133009
Adım  4   CB365471   FA474C84   C61BFEE23120
Adım  5   FA474C84   7CDDAFD2   BEF275202B2E
Adım  6   7CDDAFD2   A5773724   8F5F6A741892
Adım  7   A5773724   DB2FBDE9   EA73FD45007B
Adım  8   DB2FBDE9   59696218   9DDF4907B848
Adım  9   59696218   ED9FF14E   FF09DA58060B
Adım  10   ED9FF14E   9FCF316C   3EE2BDDE5008
Adım  11   9FCF316C   467CE9F8   9F1D1E007368
Adım  12   467CE9F8   87B8E7A4   6E2AFDB0B820
Adım  13   87B8E7A4   62BC1281   9FFC2CE00E32
Adım  14   62BC1281   07981C4E   CA2FFA1D2A1A
Adım  15   07981C4E   3843EE5A   FCFE2D355050
Adım  16   F36DF8C8   3843EE5A   BD9DA6DA0A84
Şifreli Metin :  706A189FC6DC7F4D
```

DEŞİFRELEME

23-) Bulduğumuz anahtarlarımızı tersine çeviriyoruz ve tekrardan aynı işlemler tekrar edilerek deşifreleme yapılır.

```
Şifre Çözme
İlk Permütasyon işleminden sonra : F36DF8C83843EE5A
Adım  1   3843EE5A   07981C4E   BD9DA6DA0A84
Adım  2   07981C4E   62BC1281   FCFE2D355050
Adım  3   62BC1281   87B8E7A4   CA2FFA1D2A1A
Adım  4   87B8E7A4   467CE9F8   9FFC2CE00E32
Adım  5   467CE9F8   9FCF316C   6E2AFDB0B820
Adım  6   9FCF316C   ED9FF14E   9F1D1E007368
Adım  7   ED9FF14E   59696218   3EE2BDDE5008
Adım  8   59696218   DB2FBDE9   FF09DA58060B
Adım  9   DB2FBDE9   A5773724   9DDF4907B848
Adım  10   A5773724   7CDDAFD2   EA73FD45007B
Adım  11   7CDDAFD2   FA474C84   8F5F6A741892
Adım  12   FA474C84   CB365471   BEF275202B2E
Adım  13   CB365471   C70D46B8   C61BFEE23120
Adım  14   C70D46B8   6780CF9C   BDD634133009
Adım  15   6780CF9C   F0AAF0AA   F222FF504399
Adım  16   CC00CCFF   F0AAF0AA   C3BF4A01A066
Metin :  0123456789ABCDEF
```

3-) Dijital İmza

Dijital imza dijital mesajların veya belgelerin gerçekliğini doğrulamak için imza ve matematiksel bir şemadır. Ön koşulların karşılandığı geçerli bir dijital imza, alıcıya mesajın bilinen bir gönderici tarafından oluşturulduğuna inanması için çok güçlü bir neden verir (kimlik doğrulama).

Dijital imzalar açık anahtarlı şifreleme ve asimetrik kriptografi kullanır.

1-) RSA'yı kullanarak 1024 bitlik private ve public anahtar üretimi yapılır.

2-) İmzalamak istediğimiz metin SHA256'lık bir Hash fonksiyonundan geçirilir.

3-) Oluşturduğumuz hash'i imzalamak için private anahtar kullanılır.(hash^d%n)

İmzalama gerçekleşir.

4-) İmzaladığımız hash'i tekrardan deşifrelemek için public anahtar kullanılır.(hash^e%n)

5-) Deşifrelediğimiz hash ve imzaladığımız metnin hash'i birbirine eşit olup olmadığı kontrol edilir. Eşitse imza geçerlidir. Değilse imza geçerli değildir.

```
Private Key :  ( 44556019238447546592658937383883805365696680545304070006393832409464152440066174443
Public Key :  ( 65537 , 122434709971913495221932443577676853343885255886691523522391303757612249830
Hash :  28950674917331271124690315469910439035885886896599885379524898251282567332236
Imza: 0x7d75e08a782fcee506c0e157280bd59fb170c4cd907b9ed1358eb0c3a78137db8329a9fdfcd527cced33b8a020d
Imza Geçerlidir .
----------------------------------------------------------
Geçersiz İmza
Hash :  107011848751014776079645532982668968968550171680674644898533087184517780524455
Imza Geçerli Değildir .!
```