BİL 470 DÖNEM ÖDEVİ

Programlama Projesi :

Projemi python programlama dilini kullanarak gerçekleştirdim. Ödev de bulunan bütün sorunları çözdüm. A , b , c ve d şıklarını başarılı bir şekilde tamamladım. Projeyi tek bir dosya üzerinde gerçekleştirdim. Proje çalıştırıldığında şifrelenmiş bir dosyanın gönderimi simüle edilmiştir.

a) AES şifreleme algoritmasının gerçeklenmesi ve şifreleme/deşifrelemede kullanılması(test verileri ile birlikte).

AES algoritmasının 128 bit ile çalışan modelini gerçekleştirdim. Algoritma inputlarım aşağıdaki şekildedir.

```
Plain Text :  samet gulmez 123
Key :         Thats my key for
```

Girdi olarak verilen 16 bytelık stringleri ilk olarak hexdecimal değerlere çeviriyorum ve daha matrixleirini oluşturuyorum. Bu işlemlerden sonra şifreleme döngüsü başlayacaktır. Döngüden önce son işlem olarak elimizde bulunan key ile her döngüde kullanılacak 10 adet key üretilmektedir. Bu keylerde matrix haline getirilmiştir. Bu keyler expandKey() methodu ile oluşturulmaktadır. Aşağıda verilen inputun hexedecimale çevrilip matrixe çevrilmiş hali bulunmaktadır.

```
0x73 0x74 0x6c 0x20
0x61 0x20 0x6d 0x31
0x6d 0x67 0x65 0x32
0x65 0x75 0x7a 0x33
```

Döngümüz 0. Tur ile başlar. 0. Turda üretilen keylere 0. Sı alınır ve girdi olarak verilen plaintext matrixi ilex or işlemi uygulanır. XOR işlemi add_round_key()method ile gerçeklenmektedir. Şimdi ise 9. Turun sonuna kadar bir döngü oluşturulmuştur.

Döngü:

- ➔ Xor işleminde sonra çıkan matrix sonucu S-Box 2 boyutlu listesinde ki yerinde bulunan değer ile değiştirilmektedir.
- ➔ Daha sonra çıkan matrix sonucunun her satır bulunduğu sıranın numarası kadar sola kaydırılmaktadır.
- ➔ Çıkan matrix sonucuna mix_columns() methodunda bulunan işlem uygulanmaktadır.

➔ En son çıkan sonuc döngüden önce oluşturmuş olduğumuz key listesinden kendi sırasında bulunanı alır ve XOR işlemi uygulanır.

Döngümüzün 9 işlemi tamamlandıktan sonra. 10 rounda geçiyoruz. Burada yapılan işlemin döngüde yapılandan tek farkı mix_columns işleminin yapılmamasıdır.

10. tur :

➔ Xor işleminde sonra çıkan matrix sonucu S-Box 2 boyutlu listesinde ki yerinde bulunan değer ile değiştirilmektedir.
➔ Daha sonra çıkan matrix sonucunun her satır bulunduğu sıranın numarası kadar sola kaydırılmaktadır.
➔ En son çıkan sonuc döngüden önce oluşturmuş olduğumuz key listesinden kendi sırasında bulunanı alır ve XOR işlemi uygulanır.

Bütün işlemlerin sonucunda verilen plaintext şifrelenmiş oldu. Şifrelenmiş matrix aşağıdaki şekildedir.

```
0xa4 0x50 0x66 0x9a
0xcd 0xf5 0x56 0xf2
0x76 0x75 0x80 0x76
0x56 0x83 0x4f 0xf2
```

Deşifreleme :

Şifreleme işleminde yapılan işlemlerinin tam tersi yapılarak şifrelenmiş verinin şifrelenmemiş hali elde edilir. Deşifreleme sonrası çıkan sonuç aşağıdaki gibidir.

```
0x73 0x74 0x6c 0x20
0x61 0x20 0x6d 0x31
0x6d 0x67 0x65 0x32
0x65 0x75 0x7a 0x33
```

Verilen girdinin hexadecimal şekilde çıkarılan matrix ile aynıdır. Bu yüzden şifreleme işlemi doğru bir şekilde çalışmıştır.

B) AES algoritmasının CBC ve OFB modu bu soruda gerçeklenmiştir. Bu modlar şifreleme ve deşifreleme işlemlerine çok benzemektedir. Fakat burada sadece 16 bytelık girdi ile değilde 16 bytetan daha büyük girdiler kullanılabilmektedir.

CBC

CBC kipinde her açık metin bloku şifrelenmeden önce bir önceki kapalı metin bloku ile XORlanır. Bu sayede her kapalı metin bloku kendisinden önce gelen tüm açık metinlere bağımlı olmuş olur. Bir mesajın aynı anahtar altında tekrar şifrelendiğinin anlaşılamaması için ilk blokta ilklendirme vektörü (IV) kullanılmalıdır.

```
Plain Text :  Two One Nine TwoTwo One Nine TwoSamet Sulo Samet
Key :         Thats my key for

CBC ile şifrelenmiş metin

0xdf 0x61 0xcb 0xe2 0x23 0xa7 0xf3 0xbf 0xb9 0xb6 0x02 0xec 0x15 0xe8 0x92 0x86

0x0b 0x26 0xaf 0x7d 0x89 0x0f 0xb9 0x10 0xa6 0xe3 0x98 0x6f 0x67 0x6c 0x78 0x48

0x4b 0x8f 0x12 0x72 0x08 0x16 0x6c 0x28 0x9a 0xef 0xca 0xba 0xfe 0x69 0x42 0xda


Process finished with exit code 0
```

Şifrelenmiş olarak çıkan sonuçta her satır 16 bytelık şifrelenmiş stringi temsil ediyor.

Deşifreleme işlemi için her bir şifrelenmiş metin kendinden sonraki deşifrelenmiş metin ile XOR işlemine sokulur. Ancak sadece ilk deşifreleme uygulanan metin ilklendirme vektörü denilen vektör ile XOR'lanır. Güvenlik nedeni ile ilklendirme vektörü her CBC işleminde farklı olmalıdır. Deşifreleme sonucu aşağıdaki gibidir.

```
0x54 0x77 0x6f 0x20 0x4f 0x6e 0x65 0x20 0x4e 0x69 0x6e 0x65 0x20 0x54 0x77 0x6f

0x54 0x77 0x6f 0x20 0x4f 0x6e 0x65 0x20 0x4e 0x69 0x6e 0x65 0x20 0x54 0x77 0x6f

0x53 0x61 0x6d 0x65 0x74 0x20 0x53 0x75 0x6c 0x6f 0x20 0x53 0x61 0x6d 0x65 0x74
```

OFB

OFB kipi bir blok şifreyi bir senkron akış şifre yapar. Anahtar akışı blokları oluşturur, bunlar daha sonra şifreli metin üretmek için şifresiz metin bloklarıyla XORlanır. Diğer akış şifrelerinde olduğu gibi, şifreli metinde bir bit döndürmek şifresiz metinde aynı konumda döndürülmüş bit üretir. Bu özellik, şifrelemeden önce uygulandığında bile hata düzeltme kodlarının normal çalışmasına izin verir.

```
Plain Text :  Two One Nine TwoTwo One Nine TwoSamet Sulo Samet
Key :         Thats my key for

OFB ile şifrelenmiş metin

0x37 0x1e 0xdb 0xf1 0x4f 0x34 0x50 0x54 0xaf 0xdd 0xcc 0xb7 0x5e 0xed 0xe0 0x7d

0x2b 0x31 0xf2 0xd3 0xab 0xd4 0x81 0x6e 0x6e 0x6 0xc0 0xac 0x62 0x87 0x8e 0x56

0x86 0xd5 0x1 0xe2 0x7 0xd6 0x39 0x7d 0x88 0xc7 0x47 0x3a 0x31 0x60 0xc9 0x9e
```

Deşifreleme kısmında başta şifreleme için kullanılan yöntemdeki şifrelenmek istenen metin yerine sıralı şekilde deşifrelenmek istenen metin konur ve işlem aynı sırayla ve aynı işlemlerle tekrar çalışır. Sonunda ilk başta şifrelenmek istenen metin elde edilmiş olur.

```
Plain Text :  Two One Nine TwoTwo One Nine TwoSamet Sulo Samet
Key :         Thats my key for

OFB ile Deşifrelenmiş metin

0x54 0x77 0x6f 0x20 0x4f 0x6e 0x65 0x20 0x4e 0x69 0x6e 0x65 0x20 0x54 0x77 0x6f

0x54 0x77 0x6f 0x20 0x4f 0x6e 0x65 0x20 0x4e 0x69 0x6e 0x65 0x20 0x54 0x77 0x6f

0x53 0x61 0x6d 0x65 0x74 0x20 0x53 0x75 0x6c 0x6f 0x20 0x53 0x61 0x6d 0x65 0x74
```

C)

Burada ilk olarak verilen dosyanın özütür çıkartılmıştır.

Özütünden çıkan sonuç ve çıkan sonucu hexadecimal matrix hali aşağıdaki gibidir.

```
8ea7feecc37057fb6f776c00827e9c16ae884270bacaebf0da364ae8f2e3309f
----------------HASH FıLE HEX-----------------------
0x38 0x65 0x61 0x37 0x66 0x65 0x65 0x63 0x63 0x33 0x37 0x30 0x35 0x37 0x66 0x62

0x36 0x66 0x37 0x37 0x36 0x63 0x30 0x30 0x38 0x32 0x37 0x65 0x39 0x63 0x31 0x36

0x61 0x65 0x38 0x38 0x34 0x32 0x37 0x30 0x62 0x61 0x63 0x61 0x65 0x62 0x66 0x30

0x64 0x61 0x33 0x36 0x34 0x61 0x65 0x38 0x66 0x32 0x65 0x33 0x33 0x30 0x39 0x66
```

Daha sonra özüt işleminden çıkan sonucu CBC veya OFB modu ile şifreliyoruz. Şifrelenmiş özütün sonucu aşağıdaki gibidir.

```
------------HASH ENCRYPT--------------------
0xf2 0x6f 0xe1 0x8e 0x40 0x1d 0x80 0xf4 0x86 0x9c 0x8a 0x4b 0x1f 0xe5 0x3f 0x51

0x73 0xf4 0xe7 0xc8 0xd5 0x51 0x1e 0x27 0x8e 0x5e 0x41 0xae 0x0b 0x2b 0x35 0xe3

0x97 0x02 0x9d 0xdd 0xf1 0x82 0x03 0x61 0x8e 0xfc 0x8b 0xfe 0xac 0xc2 0x7a 0x47

0x71 0x81 0xec 0x4b 0x13 0xd3 0x81 0x2e 0x76 0xfa 0xda 0x60 0x39 0x40 0xae 0x47
```

Daha sonra şifrelenen özüt gönderilmek istenen dosyanın sonuna eklenmektedir. Sonuna şifrelenmiş özütün eklenmiş olduğu dosya gönderilir. Dosyayı alan kişi ise aşağıdaki işlemleri gerçekleştirmektedir.

D)

Dosyayı deşifreleme işlemi gerçekleştirilecektir. İlk olarak dosyanın sonun şifrelenmiş özüt alınır ve deşifrelenir. Bu sonuc aşağıda gösterilmiştir.

```
0x38 0x65 0x61 0x37 0x66 0x65 0x65 0x63 0x63 0x33 0x37 0x30 0x35 0x37 0x66 0x62

0x36 0x66 0x37 0x37 0x36 0x63 0x30 0x30 0x38 0x32 0x37 0x65 0x39 0x63 0x31 0x36

0x61 0x65 0x38 0x38 0x34 0x32 0x37 0x30 0x62 0x61 0x63 0x61 0x65 0x62 0x66 0x30

0x64 0x61 0x33 0x36 0x34 0x61 0x65 0x38 0x66 0x32 0x65 0x33 0x33 0x30 0x39 0x66
```

Sonuç olarak dosyanın sonuna eklenmiş şifrelenmiş özüt kaldırılmış oldu. Dosyanın kendisi kaldı. Daha sonra elimizde kalan dosyanın özütünü alıyoruz. Böylelikle gönderilen dosyanın özütü ile dosyanın sonuna eklenmiş özütü karşılaştırıyoruz. Gönderilen dosyanıın özütü ise aşağıdaki şekildedir.

```
Gönderilen dosyanın sonunda hash çıkartıldıktan sonra kalan halinin alınmış Hashi

0x38 0x65 0x61 0x37 0x66 0x65 0x65 0x63 0x63 0x33 0x37 0x30 0x35 0x37 0x66 0x62

0x36 0x66 0x37 0x37 0x36 0x63 0x30 0x30 0x38 0x32 0x37 0x65 0x39 0x63 0x31 0x36

0x61 0x65 0x38 0x38 0x34 0x32 0x37 0x30 0x62 0x61 0x63 0x61 0x65 0x62 0x66 0x30

0x64 0x61 0x33 0x36 0x34 0x61 0x65 0x38 0x66 0x32 0x65 0x33 0x33 0x30 0x39 0x66
```

Elimizde bulunan bu iki adet özütü alınmış dosya karşılaştırılır. Eğer bir farklılık yok ise dosya değiştirilmemiş demektir. Eğer farklılık var ise değiştirildiği anlamına gelmektedir.

```
Gönderilen dosyanın sonuna eklenmiş olan Hash

0x38 0x65 0x61 0x37 0x66 0x65 0x65 0x63 0x63 0x33 0x37 0x30 0x35 0x37 0x66 0x62

0x36 0x66 0x37 0x37 0x36 0x63 0x30 0x30 0x38 0x32 0x37 0x65 0x39 0x63 0x31 0x36

0x61 0x65 0x38 0x38 0x34 0x32 0x37 0x30 0x62 0x61 0x63 0x61 0x65 0x62 0x66 0x30

0x64 0x61 0x33 0x36 0x34 0x61 0x65 0x38 0x66 0x32 0x65 0x33 0x33 0x30 0x39 0x66

Dosya üzerinde değşiklik yapılmamıştır !!!!
```

Bu değişikliği test edebilmek için program içinde dosya gönderilmeden önce içinde bir değişklik yaptım. Böylelikle dosya içinde deişiklik olduğunu bulmamız gerekiyor.

```
Gönderilen dosyanın sonunda hash çıkartıldıktan sonra kalan halinin alınmış Hashi

0x38 0x65 0x61 0x37 0x66 0x65 0x65 0x63 0x63 0x33 0x37 0x30 0x35 0x37 0x66 0x62

0x36 0x66 0x37 0x37 0x36 0x63 0x30 0x30 0x38 0x32 0x37 0x65 0x39 0x63 0x31 0x36

0x61 0x65 0x38 0x38 0x34 0x32 0x37 0x30 0x62 0x61 0x63 0x61 0x65 0x62 0x66 0x30

0x64 0x61 0x33 0x36 0x34 0x61 0x65 0x38 0x66 0x32 0x65 0x33 0x33 0x30 0x39 0x66

Gönderilen dosyanın sonuna eklenmiş olan Hash

5 0x65 0x61 0x37 0x66 0x65 0x65 0x63 0x63 0x33 0x37 0x30 0x35 0x37 0x66 0x62

0x36 0x66 0x37 0x37 0x36 0x63 0x30 0x30 0x38 0x32 0x37 0x65 0x39 0x63 0x31 0x36

0x61 0x65 0x38 0x38 0x34 0x32 0x37 0x30 0x62 0x61 0x63 0x61 0x65 0x62 0x66 0x30

0x64 0x61 0x33 0x36 0x34 0x61 0x65 0x38 0x66 0x32 0x65 0x33 0x33 0x30 0x39 0x66

Dosya üzerinden değişiklik yapılmıştır !!!!
```

Yazmış olduğum programın simülasyonu bir dosyanın gönderilmesi ve gönderilen dosyada değişiklik yapılıp yapılmadığı bulunuyor. Simülasyonda şifreleme/deşifreleme işlemi , CBC ile şifreleme/deşifreleme işlemi ve OFB ile şifreleme / deşifreleme işlemi ekstra olarka gösterilmemektedir. Dosya gönderme işlemi içerisinde bu işlemler halli hazırda kullanılmaktadır.