**Crypto 101**

**Belgeler:**

- https://www.mehmetince.net/crypto-101-1-merhaba-exclusive-or-xor/
- https://www.mehmetince.net/crypto-101-2-block-cipher-encryption-ve-des-analizi/
- https://www.mehmetince.net/crypto-101-3-dese-yonelik-saldirilar-3des-aes-metodlari/
- https://www.mehmetince.net/crypto-101-4-stream-cipher-ve-random-number-generation/
- https://www.mehmetince.net/crypto-101-5-sifreleme-operasyonu-modlari-ecb-cbc-ofb/

**Notlar:**

a)

**XOR Operatörü / Kapısı**

Kriptolojiyi anlamak için XOR operatörü / kapısı önemlidir.

b)

**XOR Operatörü / Kapısı Nasıl Çalışır?**

XOR (⊕) operatörü şu şekilde işler:

$0 \oplus 0 = 0$

$0 \oplus 1 = 1$

$1 \oplus 0 = 1$

$1 \oplus 1 = 0$

Yani

- “$0 \oplus 0$” ve “$1 \oplus 1$” durumlarında 0 çıkışı,
- “$0 \oplus 1$” ve “$1 \oplus 0$” durumlarında 1 çıkışı

vermektedir.

c)

**XOR Operatörü / Kapısı ile Şifreleme**

- **Key (Anahtar) Belirleme**

  Birbirleri ile konuşmak isteyen Alice ve Bob'un aralarında gizli bir ortak anahtar belirlemiş olduklarını var sayalım.

  Anahtar = 10111

- **Gönderilecek Plain-Text Mesaj**

  Alice'in göndermek istediği mesaj şu olsun:

  Mesaj = 10101

- **Mesajı Şifreleme**

  Alice mesajını Bob'a göndermeden önce anahtar ile şifreleme işlemini gerçekleştirecektir. Bunun için şu adımı uygular.

  Plain-Text Mesaj⊕ Anahtar = Şifrelenmiş Mesaj

  Yani şu işlem uygulanır:

  10101 ⊕ 10111 = 00010

  (plain-text mesaj) ( anahtar) = ( şifrelenmiş mesaj )

  Göndermek istediği orjinal mesaj 10101 iken Alice bunun yerine anahtarla şifrelenmiş hali olan 00010 mesajını gönderecektir. Bu sayede mesajı taşıyacak olan kişiden orijinal mesaj gizlenmiş olacaktır.

- **Mesajı Deşifreleme**

  Mesajı alan Bob ise Alice'in gönderdiği şifreli mesajı aynı anahtar ile xor'layarak deşifreleyecektir:

  Gelen şifrelenmiş mesaj ⊕ Anahtar = Plain-Text Mesaj

  Yani şu işlem uygulanır:

  00010 ⊕ 10111 = 10101

  ( şifrelenmiş mesaj) ( anahtar ) = (plain-text mesaj)

- **Sonuç**

  Exclusive OR yani XOR bizim için kriptolojinin temel matematik operatörüdür. Plain-text veri belirlenen anahtar ile xor'lanarak şifrelenmiştir ve yine aynı anahtar ile xor'lanarak deşifrelenmiştir. Arada trafiği dinleyen bir saldırgan ise şifrelenmiş mesajı elde etse bile

şifreleme ve deşifrelemede kullanılan key'i (anahtarı) bilmediğinden mesaj gizliliği temin edilmiş olacaktır.

d)

**Simetrik Anahtar ile Şifreleme Nedir?**

Simetrik Anahtar ile Şifreleme (symmetric-key encryption) bir plain-text verinin Alice ve Bob arasında gidip gelmesi süresince **aynı** anahtarın kullanılıyor olması anlamına gelir. Diğer bir deyişle şifreleme için kullanılan anahtar aynı zamanda deşifreleme işleminde de kullanılmaktadır. İletişimde gizlilik bu aynı anahtarın Alice ve Bob harici kimse tarafından bilinmiyor olmasıyla ayakta durur.

e)

**XOR Operatörünün / Kapısının Özellikleri**

1. XOR Değişme Özelliği

   XOR işleminin, a ⊕ b = b ⊕ a eşitliği vardır. Buna orta okul yıllarında öğrenilen XOR'un değişme özelliği denir.

2. XOR Sıfırlama Özelliği

   Bir değişken kendisiyle XOR işlemine girdiğinde sonuç 0'dır. Yani; a ⊕ a = 0

3. XOR ile Sıfır İşlemi Özelliği

   Herhangi bir değişken sıfır ile XOR işlemine girerse sonuç kendisine eşittir. Yani; a ⊕ 0 = a

Tüm bu kurallar bir arada kullanılacak olursa a ⊕ b ⊕ a = b olduğunu ispatlayalım.

Soru :

- a ⊕ b ⊕ a =? b // a ⊕ b ⊕ a ifadesi b'ye eşit midir?

Cevap:

- a ⊕ b ⊕ a = a ⊕ a ⊕ b // 1. kural, değişme özelliği
- a ⊕ a ⊕ b = 0 ⊕ b // 2. kural sıfırlama özelliği
- 0 ⊕ b = b // 3. kural, 0 ile işlem sonucu özelliği

Yani a ⊕ b ⊕ a ifadesinin b değerine eşit olduğunu görüyoruz.

f)

**One-Time Pads Nedir?**

One-time pad 1918 yılında Gilbert Vernam tarafından bulunmuş bir yöntemdir. Kabaca şifreli iletişimde tek kullanımlık olarak üretilen bir anahtarı ifade eder. Eğer sadece bir defa kullanılmak üzere üretilen anahtarda bitler gerçekten rastgele ise ve üretilen bu anahtar sadece ve sadece 1 defa kullanılıyor ise ciphertext'i elde eden saldırganın plaintext'e ulaşması mümkün değildir.

g)

**One-Time Pads (Birden Fazla Anahtar) Neden Gereklidir?**

One-Time Pads kullanılmadığında - yani simetrik şifrelemede "tek kullanımlık" birden fazla anahtar kullanılmadığında - ciphertext örneklerini elde eden saldırgan ciphertext örnekleri üzerinden anahtarı bilmeden plaintext veriye ulaşabilir.

Örneğin Alice ile Bob konuşmalarını simetrik şifreleme ile (yani aynı anahtar ile) ve tek kullanımlık olmayan (defalarca kullanılabilen) anahtar ile yapıyor olsunlar. Eve isimli bir saldırgan ise arada iletişimi dinliyor olsun. Bu durumda her defasında kullanılan aynı anahtar ile şifrelenmiş iki ayrı mesajı sniff'leme ile elde eden saldırgan bu iki şifrelenmiş mesajı birbiriyle XOR'layarak plain-text mesajı elde edebilecektir ve bunu anahtarı bilmeden yapabilecektir. P.O.C şu şekildedir:

**Değişkenler:**

C1 : CipherText #1 // Şifrelenmiş Mesaj #1
C2 : CipherText #2 // Şifrelenmiş Mesaj #2

P1 : PlainText #1 // Açık Metin Mesaj #1
P2 : PlainText #2 // Açık Metin Mesaj #2

K : Key // Anahtar

**İspat:**

Sniff'lenen şifrelenmiş iki mesaj (C1 ve C2) XOR işlemine tabi tutulur.

$C1 \oplus C2 = ( P1 \oplus K ) \oplus ( P2 \oplus K )$ // Denklem esas parçalarına ayrılır.

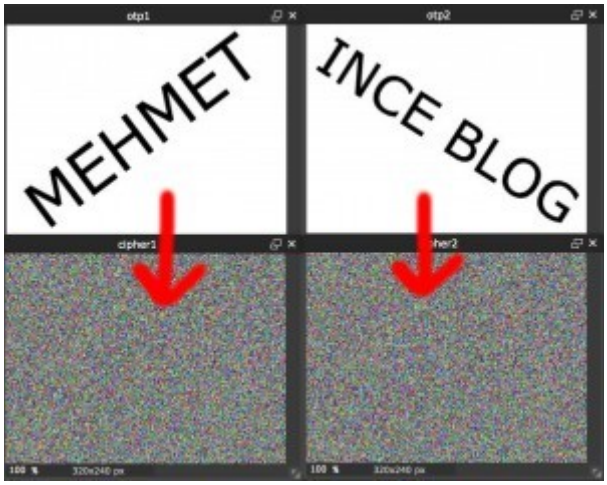
$= P1 \oplus P2 \oplus K \oplus K$ // XOR değişme özelliği kullanılır ve // sıralama değiştirilir.

$= P1 \oplus P2 \oplus 0$ // XOR sıfırlama özelliği ile K elimine // edilir. Böylece bilinmesine gerek yok.

$= P1 \oplus P2$ // XOR sıfır ile işlem özelliği ile P1 ve // P2 kalır.

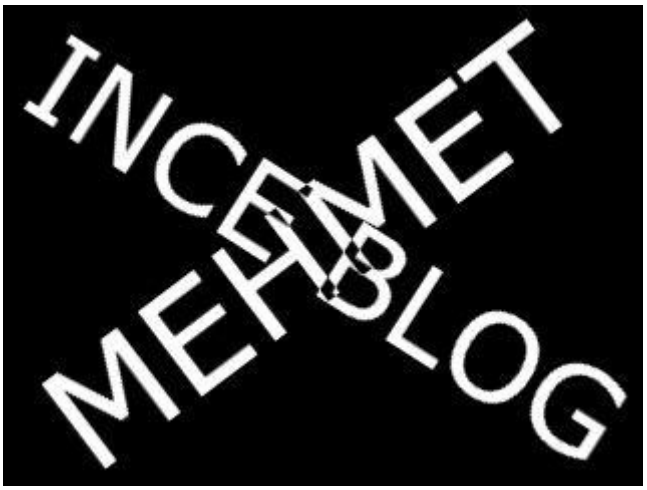
Böylece saldırgan şifrelenmiş C1 ve C2 mesajlarını XOR işlemine tabi tutarak plain text P1 ve P2 mesajlarının birleşimini elde etmiş olmaktadır.

Bu plaintext mesajların birleşimi durumunu şöyle izah edebiliriz. Örneğin iki ayrı resim (solda MEHMET, sağda INCE BLOG) olsun.




Bir python script’i ile soldaki resmi ve sağdaki resmi her defasında kullanılan aynı bir anahtar ile şifreledik diyelim ve iki ayrı şifrelenmiş mesaj (iki ayrı karıncalı görüntü) elde ettik diyelim.

Bu şifrelenmiş iki resim (iki ayrı karıncalı görüntü) birbirleriyle bitsel olarak XOR işlemine tabi tutulduklarında açık metin resimlerin birleşimi elde edilecektir.




Sonuç olarak şifrelenmiş iki ayrı örnek mesaj XOR’a tabi olduğunda iki mesajın deşifrelenmiş birleşim hali elde edilir ki bu iletişimde gizliliğin kalkması anlamına gelir.

h)

**Random Number Generator (RNG)**

Random number generator fizik modellemelerinde, mühendislikte, matematiksel bilgisayar çalışmalarında, kriptografide, oyun sunucularında v.b. birçok alanda kullanılır ve önemlidir.

RNG kabaca iki türdür:

a) True Random Number Generator ( TRNG)

Mouse hareketleri, klavyede tuşlanan tuşların arasındaki gecikme süreleri v.b. parametrelerle oluşturulan random number generator'lara true random number generator denir. Örnek vermek gerekirse https://www.bitaddress.org adresi mouse hareketlerinizin koordinatlarını kaynak olarak kullanarak bitcoin adresi üretmektedir. Saldırganları bir kenara bırakırsak kendi kendinize iki kere aynı adresi üretmeniz bile neredeyse imkansızdır.

b) Pseudo Random Number Generator (PRNG)

Matematiksel bir fonksiyon ile bilgisayar sistemleri tarafından üretilen rastgele sayırlardır. Fiziksel sistemler üzerinden elde edilen TRNG'e göre gerçek rastgelelik ortadan kalktığı için (matematiksel fonksiyon ile hesaplanan bir değer oluştuğundan ötürü) adı Pseudo yani sözde rastgele sayı olarak nitelendirilmiştir.

Ek Kaynak:

https://en.wikipedia.org/wiki/List_of_random_number_generators
https://rust-random.github.io/book/guide-gen.html