İSÜ | İSTİNYE ÜNİVERSİTESİ İSTANBUL

BIL 008

BIL 008 : Kriptografi Temelleri

17/11/2020

Şifreleme/Deşifrasyon

Sonbahar 2020

Ahmad Al Khas

1 Giriş:

Bu proje ödevi, kriptografinin Şifreleme / Şifre Çözme algoritmalarının temellerini kavramak için sağlanmıştır. Bu gerçek bir kriptografik algoritma değildir, ancak bazı temel algoritmaların yapısını simüle eder. Algoritmayı Python veya MATLAB programlama dillerini kullanarak oluşturmanız istenir. Algoritma 2. bölümde ayrıntılı olarak açıklanmıştır. 2.1,2.2, 2.3 ve 2.4 bölümlerinde açıklanan adımları kullanarak bir kriptografik şifreleme sistemi oluşturacaksınız. Sisteminiz, bir resim veya metin mesajının kullanıcı girdilerini kabul edecektir. Sistemin girdisi düz metin olarak kabul edilir.

Bir resim veya bir metin mesajından oluşan *düz metni* şifreleyebilmek için, önce sisteminizin veri türünü kontrol etmesi gerekir. *Metnin* türünü onayladıktan sonra, mesajı ikili bit akışına dönüştürerek dijitize etmeniz gerekecektir **(Bu görev için internet kaynaklarını kullanabilirsiniz veya ALMS'ye yüklenen uygulama derslerinin kodlarını kullanabilirsiniz)**. Sistemde 2 karıştırma anahtarı ve bir ana anahtar var. Her bir anahtar şekil ve yapı 2. bölümde tam olarak açıklanmıştır.

2 Şifreleme:

Şifreleme / şifre çözme algoritması bir seferde *düz metnin* 256 bitini kabul eder, bu da açık *metinlerinizin* 256 bitlik bloklara bölünmesi gerektiği anlamına gelir. Algoritmanız sabit blok boyutlarında çalışacağından, *düz metin* akışı boyutunuz bölünmeden önce 256 bitin katları olmalıdır. Bunu onaylamak için, son bloğun başına sıfırlar ekleyebilirsiniz **(256 bitten az olması durumunda)**. Örneğin, *metninizin* ikili bit akışı 500 bit'e eşitse, $500 \leq 2 * 256(512)$ 'den bu yana iki bloğa bölünecektir, bu nedenle ilk blok 256 bit ve ikincisi Sıfırlar (12) + 244 bit. Başka bir örnek, eğer *düz metin* 978 ise, o zaman $978 \leq 4 * 256(1024)$ 'den beri 4 bloğa bölünecek, bu nedenle ilk üçü 256 bit olacak ve son blok Sıfırlar (46) + 210-bit olacaktır. Bu, *düz metin* ikili akışının herhangi bir boyutunda uygulanabilir.

Algoritma daha sonra blokları teker teker alacak ve aşağıdaki işlemleri gerçekleştirecektir: Sisteminizin 256 bitlik blok boyutlarında çalışacağını unutmayın, aşağıdaki örnekler, sadece gösterim amacıyla 32 bitlik blok boyutları kullanılarak yapılmıştır.

2.1 Bit çevirme:

Bu, sadece ikili akışın bitlerinin konumlandırma sırasını değiştirmekten oluşan oldukça basit bir işlemdir. Birincisi son olacak ve sonuncusu birinci olacak, ikincisi sondan ikincisi olacak ve tam tersi. Bu, tüm bitler çevrilene kadar devam edecektir. Figure 1, 32 bitlik ikili akış çevirme işleminin bir örneğini gösterir. Figure 2, ters çevrilmiş ikili akışın sonucunu göstermektedir.

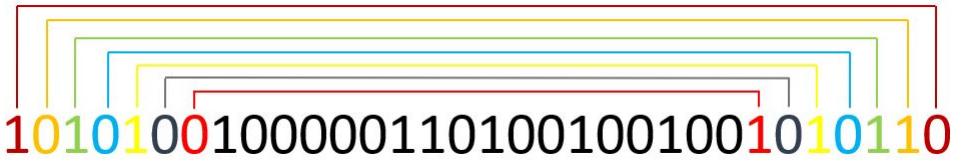

Figure 1: 32 bit ikili akışın bit çevirme işlemi örneği

Figure 2: Tüm bitleri olan ikili akış ters çevrilir

2.2 İkili akışı karıştırma:

Bu bölümde ikili akışı karıştırmak için ilk anahtarı kullanacaksınız. Anahtar dört sayıdan oluşan bir diziden oluşacaktır, sayı aralığı [1-4] arasında olacaktır. Her numara dizide bir kez kullanılacak ve konumları rastgele belirlenecektir. Daha sonra ikili akışı her biri 64 bitlik dört eşit bloğa ayıracaksınız. Şimdi blokların sırasını anahtar sıra numarasına göre değiştireceksiniz.
Figure 3, 32 bitlik ikili akış karıştırma işleminin bir örneğini göstermektedir, $anahtar1 = [1, 4, 3, 2]$, böylece blokların sırası yeniden düzenlenecektir. anahtardaki sayıların sırasına göre. Böylece, ilk blok önce sonra dördüncü, üçüncü ve son olarak ikinci bloğa gidecektir.

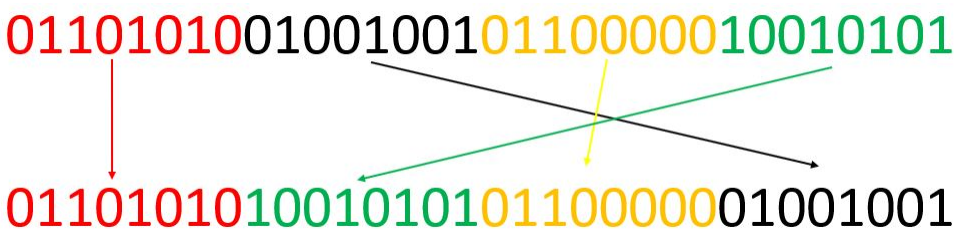

Figure 3: İkili akış, ilk anahtarı kullanarak karıştırma işlemini engeller

2.3 XOR:

Bu bölümde sistemin ikinci anahtarını kullanacaksınız. Anahtar, 128 bitlik bir ikili akış olacaktır. Bu anahtar, bir zaman pedi şifreleme algoritmasında yapıldığı gibi yalnızca bir kez kullanılacaktır. 128 bitlik anahtarı aldıktan sonra, başka bir 128 bit ikili akış oluşturmak için tüm bitlerini ters çevireceksiniz. Son olarak, anahtarı XOR ve bölüm 2.2 ile elde edilen *düz metnin* ikili akışı ile tersini yapacaksınız. Orijinal anahtar *düz metnin* ilk 128 biti ile XOR işlemi olacak ve ters çevrilen anahtar son 128 bit ile XOR işlemi yapılmış olacaktır.
Figure 4, 16 bitlik bir anahtarın (gri renkli) ve ters çevrilmiş 16 bitlik (kahverengi renkli) bir anahtarını göstermektedir. Bundan sonra, figure 5'te gösterildiği gibi *düz metnin* ikili akışı ile XOR işlemi yapılır.

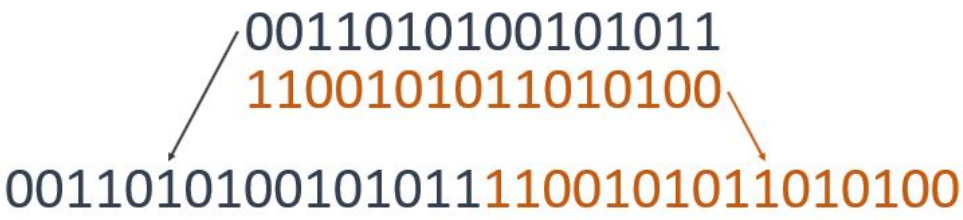

Figure 4: Anahtarın ve onun tersinin oluşturulması

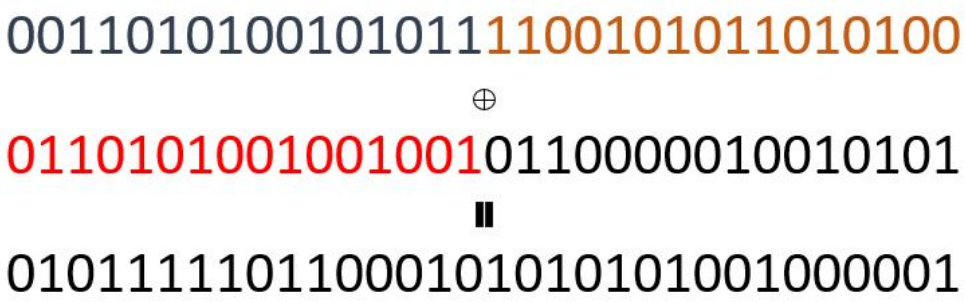

Figure 5: Oluşturulan anahtarı ikili akışla XORlamak

2.4 İkinci ikili akışı karıştırma:

Bu, şifrelemenin son adımıdır. Bölüm 2.2'de belirtilen karıştırma adımına benzer. Ancak, anahtar diziniz dört yerine sekiz sayıdan oluşacak ve aralık [1-8] olacaktır. Numaraların sırası rastgeledir ve her numara bir kez kullanılacaktır. Daha sonra, bölüm 2.3'te elde edilen *düz metin* ikili akışı, her biri 32 bitlik 8 bloğa bölünecektir. Son olarak, bölüm 2.2'de yaptığınız gibi üçüncü anahtara göre onları yeniden düzenleyeceksiniz. Elde edilen ikili akış *şifreli metin* olacaktır.
Figure 6, 32 bitlik bir ikili akış örneğini $anahtar3 = [2, 3, 1, 4, 5, 8, 7, 6]$ ile göstermektedir.

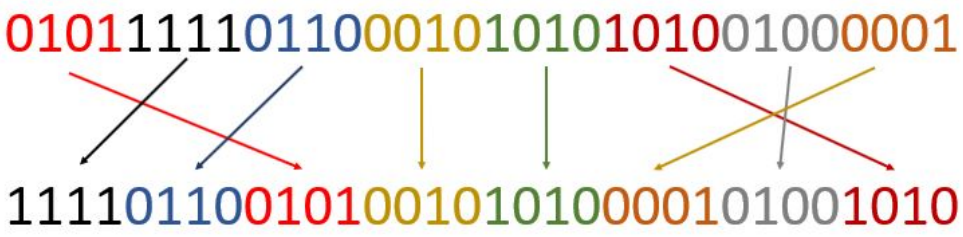

Figure 6: Üçüncü anahtarı kullanarak ikili akışı karıştırma işlemi

3 Deşifrasyon:

Bölüm 2.4'ten *şifreli metni* aldıktan sonra, şifre metnini tekrar elde etmek için şifresini çözmeniz gerekecek. Alıcının şifrelemede kullanılan üç gizli anahtara zaten sahip olduğunu unutmamalısınız. Bu nedenle, *düz metni* geri almak için şifrelemede yapılan tüm adımları tersine çevirmeniz gerekecektir.

1. Bölüm 2.4'te yapılan son işlemi tersine çevirerek başlayacaksınız, bu adımda ikili akış bloklarını orijinal sıralarına göre yeniden düzenlemeniz gerekecek. **Alıcının aynı üçüncü anahtara sahip olduğunu unutmayın.**
2. İkili akışı tersine karıştırdıktan sonra, bölüm 2.3'te yapılan aynı işlemi tekrar edeceksiniz. **İkili akışı aynı anahtarla iki kez XOR etmenin ikili akış üzerindeki anahtar etkisini gerçekten iptal ettiğini unutmayın.**
3. Ardından, bölüm 2.2'de yapılan karıştırmayı tersine çeviriniz gerekecek.
4. Son olarak, bölüm 2.1'deki gibi ikili akım bitlerini bir kez daha çevirmeniz gerekir.

Son olarak, *şifreli metnin* şifresini çözerek orijinal *düz metni* aldınız. **Görüntüyü görüntüleyin ve sonuçları karşılaştırmak için metni yazdırın.**

4 Sorular:

Lütfen uyguladığınız algoritmaya ilişkin aşağıdaki soruları yanıtlayın.

1. Bu sistemin ait olduğu çalışma modunun adı nedir?
2. Sistemin savunmasız kalmasına neden olabilecek saldırı türleri nelerdir?
3. Algoritma size göre standartlaştırılacak kadar güvenli mi? Neden?
4. Algoritma, güvenliğin temel hedeflerini sağlıyor mu? **(Gizlilik, Bütünlük, Erişilebilirlik)**
5. *Şifreli metnin* şifresini çözdükten sonra orijinal *düz metni* nasıl elde edebiliriz? çözülmez ise sebebi nedir?
6. Algoritmanın zayıf yönleri nelerdir? Bunları nasıl çözülmelidir?
7. Lütfen şifrelenmiş fotoğrafı görüntüleyin ve orijinal olanlarla karşılaştırmak için şifreli metni yazdırın.

5 Proje kuralları:

- 4 kişilik gruplar oluşturabilirsiniz.
- Her grup bir lider belirlemelidir. Grup lideri, proje ilişkin soruları sormaktan sorumludur.
- **Proje son teslim tarihi Salı gunü 10:00'dur.**
- Tüm ekip üyeleri kodun akışını çok iyi bilmelidir, bire bir görüşmeler gerçekleşebilir.
- Proje kodlarınızı .rar uzantılı sıkıştırılmış dosya halinde gönderilmelidir.
- Proje klasörüne kodun akışını kısaca açıklayan ve grup üyelerinin adını ve soyadını belirten bir README.txt dosyası eklenmelidir.