**T.C.**

**SAKARYA ÜNİVERSİTESİ**

**BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**

KRİPTOLOJİYE GİRİŞ

RC4 Algoritması Şifreleme ve Çözme İşlemi

G181210037 – Burak Koray Aksoy

|  |  |  |
| --- | --- | --- |
| Bölüm  Dersin Ögr. Görevlisi | :  : | BİLGİSAYAR MÜHENDİSLİĞİ  Doç.Dr. ÜNAL ÇAVUŞOĞLU |

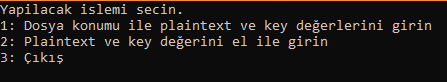
RC4 algoritması 1984 yılında Ronald RIVEST tarafından RSA veri güvenliği için tasarlanan bir akış şifreleme modelidir.

SSL/TSL ve IEEE802.11 sandartını kullanan haberleşme sistemlerinde kullanılmaktadır. IEEE802.11 wireless haberleşme standartıdır.

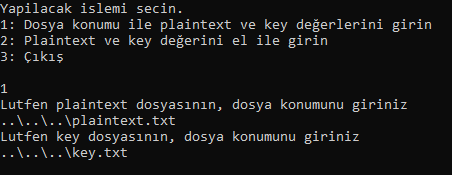
Akış şifreleme modeli olduğundan veriler bayt - bayt şifrelenir. Elimizde bir açık metin ve bir anahtar vardır, anahtarın uzunluğu açık metin kadardır. Anahtarı bir dizi olarak düşünebiliriz, 0 dan başlayarak 255’e kadar indisleri olan bir s dizisi (s[0], s[1], s[2], …, s[255]) olarak düşünebiliriz. Her dizi elamanı 1 baytlık tam sayı boyutundadır. Eğer verilen anahtar 256 bayt uzunluğunda ise hiçbir işleme tabi tutmadan şifreleme işleminde verilen anahtarı kullanabiliriz, eğer anahtar 256 bayt uzunluğundan daha kısa ise bu durumda anahtarı tekrar ederek ilgili s dizisine yerleştirilebilir. Ben uygulamamda 256 bayta kadar uzatmadım bunun yerine verilen anahtar uzunluğuna göre modulo işlemi yaptım, iki işlem de aynı sonuca varıyor.

Algoritmanın başlangıç aşaması

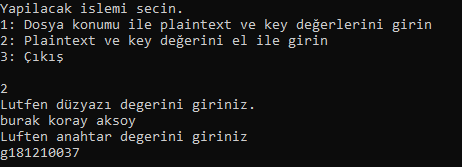
İlk olarak kullanıcın karşısına küçük bir menü geliyor. Kullanıcı düz yazı ve anahtar verilerini iki farklı şekilde girebilme seçeneğine sahip.



İlk seçenekte kullanıcıdan düzyazı ve anahtar verilerini text dosyası olarak alabilme seçeneği sunuyorum. Kullanıcı ilgili dosyanın konumunu doğru olarak girmek zorundadır. İlgili dosyanın konumu exe dosyasının konumuna göre girilmelidir örneğin exe dosyası ve text dosyaları aynı dosyanın içinde ise “plaintext.txt veya .\plaintext.txt ve key.txt veya .\key.txt” olarak girilmelidir. Kullanıcı dosya konumunu girerken 3 hata hakkına sahiptir 3. hata durumunda kullanıcı menüye geri döndürülür.

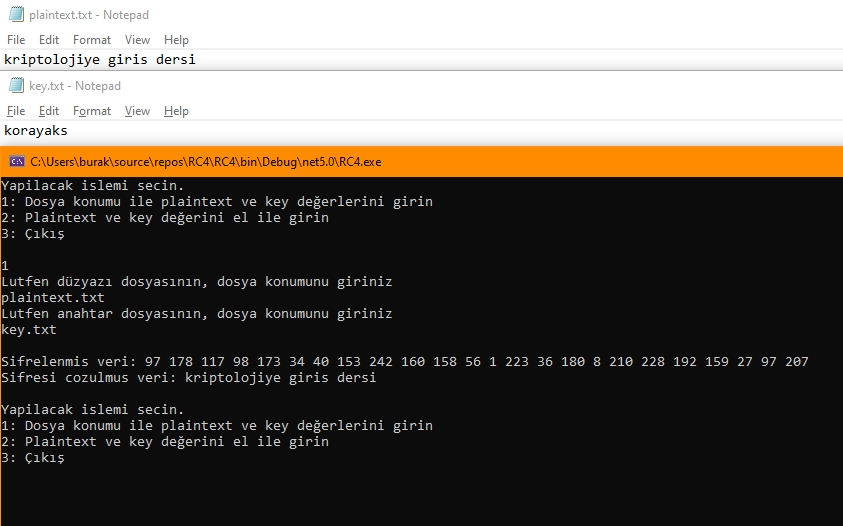


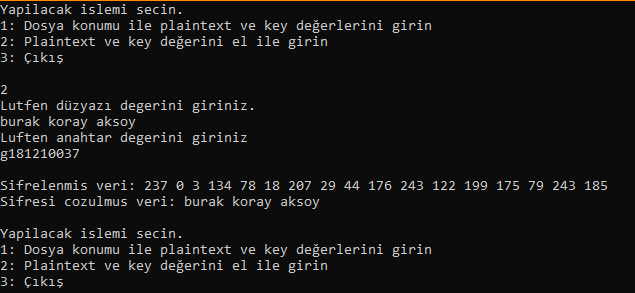
İkinci seçenekte kullanıcıdan düzyazı ve anahtar verilerini el ile girebilme seçeneği sunuyorum. Kullanıcıdan ilk olarak düzyazı verisini sonrasında da anahtar verisini alıyorum.



Kullanıcı bu iki işlemden birini seçip düzyazı ve anahtar verilerini aldıktan sonra program sırasıyla şifreleme ve şifre çözme işlemlerini gerçekleştirir. Sırasıyla, şifreleme işleminde girilen düzyazı ve anahtar verilerini ilgili şifreleme fonksiyonuna parametre olarak vererek şifrelenmiş metin elde edilir ve elde edilen metin ekrana yazdırılır. Şifrelenmiş metin ascii tablosundaki decimal formdadır. Şifre çözme işleminde ise oluşan şifrelenmiş metin ile anahtar, şifrelemede kullanılan aynı fonksiyona parametre olarak verilir ve şifresi çözülmüş düzyazı elde edilir ve elde edilen şifresi çözülmüş düzyazı ekrana yazdırılır. RC4 algoritmasında şifreleme, şifre çözmenin tersi olduğundan dolayı şifreleme ve şifre çözme işlemlerinde aynı fonksiyonu kullanıyorum.

Kullanıcı şifrelenmiş metin ve şifresi çözülmüş metnin çıktısını gördükten sonra ana menü tekrar ekrana gelir ve kullanıcıdan yeni bir işlem bekler.





Şifreleme işleminde ilk olarak kullanıcıdan alınan düzyazı ve anahtar verilerinin türlerini stringden byte dizisine dönüştürüyorum. Sonrasında bu iki veriyi şifreleme ve şifre çözme işlemi için kullandığım SifreleVeSifreCoz fonksiyonuna parametre olarak veriyorum. Bu fonksiyonda ilk olarak 256 indis uzunluğunda yeni bir s byte dizisi oluşturuluyor. Oluşturulan s byte dizisinin içini bir for döngüsü içinde dolduruyorum, 0’dan başlayarak 255’e kadar tüm indisler, indis sayısının verisi ile doldurulur. Sonrasında yeni bir for döngüsü oluşturuyorum, i ve j değerleri 0’dan başlıyor, i değişkeninin değeri her döngüde 1 artıyor ve 255’e kadar for döngüsü sürüyor. 0 değerinden başlayan j değerine yeni bir değer atanıyor bu değer (j + anahtar[i % anahtar.Length] + s[i]) değerinin 256’ya göre modunun alınması ile bulunuyor. Burada anahtar dizisinin ve yeni oluşan s dizisinin verilerinin kullanıldığını görebilirsiniz. Anahtar dizisinin i’inci indisinindeki değeri, s dizisinin i’inci değeri ve j’nin değeri toplandığında yeni j değeri elde ediliyor. Dikkat edilmesi gereken bir nokta “anahtar[i % anahtar.Length]” kısmındaki “ i % anahtar.Length ” kısmı, ilgili kısmı açıklamadan önce dökümanın en başındaki paragrafta bahsettiğim üzere anahtar dizisinin uzunluğu 256’dan küçük ise anahtar dizisi 256 byte’a uzatılmalıdır demiştim. “ i % anahtar.Length ” bu işlem ile anahtar dizimin uzunluğunu 256 byte’a kadar uzatmama gerek kalmıyor çünkü anahtar dizisinin boyutunu büyütmek için anahtar dizisinin tekrarlanıp dizinin uzunluğunun büyütülmesi gerekiyor, yaptığım işlem ile anahtar dizisinin baştaki boyutuna göre moduna alarak tekrar işlemine gerek kalmadan işlemi yapabiliyorum. Yeni j değerini elde ettikten sonra YerDegistir fonksiyonu ile s dizisindeki s[i] ve s[j] elemanlarının yerlerini değiştiriyorum.

Yukarıda anlattığım for döngüsünden sonra tamamen karıştırılmış bir s dizisi elde etmiş oluyorum. Bundan sonra şifreleme kısmına geçmiş oluyorum, ilk olarak düzyazı byte dizisinin uzunluğu kadar dönen bir döngü oluşturuyorum. Bu döngü her döndüğünde yeni bir byte elde edilmiş oluyor, döngüden elde edilen byte’lar şifrelenmiş metni oluşturmuş oluyor. Döngünün içinde yeni i ve j değerleri oluşturuluyor. İ ve j değerleri 0’dan başlayarak “i <- ((i + 1)mod 256)” ve “j <- ((j + s[i])mod 256)” formülleri sayesinde yeni değerleri oluşturuluyor sonrasında YerDegistir fonksiyonu ile birlikte s[i] ve s[j] değerleri yer değiştiriliyor. Bu işlemlerden sonra XOR işlemi yapılıp şifrelenmiş byte oluşturuluyor.

( d XOR s[((s[i] + s[j]) mod256)] ) formülü ile yeni bir byte elde ediyorum, bu byte şifrelenmiş metnin ait bir byte. Formüldeki d harfi düzyazıya ait olan bir byte’ı işaret ediyor . Şifre çözme işlemlerinde de daha önce de bahsettiğim gibi aynı işlemler yapılıyor ve sonuç olarak şifresi çözülmüş düzyazı elde edilmiş olunuyor.