T.C.

BİLECİK ŞEYH EDEBALİ ÜNİVERSİTESİ

MÜHENDİSLİK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

RC4 ALGORİTMASI İLE METİN VE GÖRÜNTÜ ŞİFRELEME

Fırat UÇAR

BİTİRME ÇALIŞMASI

DANIŞMANI : Doç. Dr. Cihan KARAKUZU

BİLECİK

3 Ocak 2018

T.C.

BİLECİK ŞEYH EDEBALİ ÜNİVERSİTESİ

MÜHENDİSLİK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

RC4 ALGORİTMASI İLE METİN VE GÖRÜNTÜ ŞİFRELEME

Fırat UÇAR

BİTİRME ÇALIŞMASI

DANIŞMANI : Doç. Dr. Cihan KARAKUZU

BİLECİK

3 Ocak 2018

BİLDİRİM

Bu kitaptaki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

İmza

Öğrencinin Adı SOYADI

Tarih:

ÖZET

BİTİRME ÇALIŞMASI

RC4 ALGORİTMASI İLE METİN VE GÖRÜNTÜ ŞİFRELEME

Fırat UÇAR

Bilecik Şeyh Edebali Üniversitesi
Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü

Danışman: Doç. Dr. Cihan KARAKUZU

2018, 42 Sayfa

Jüri Üyeleri	**İmza**
.....	.....
.....	.....
.....	.....

RC4 , 1987 yılında Ron Rivest tarafından geliştirildi. O zamandan beri çeşitli uygulamalarda kullanılmaktadır. ARC4 ve ya ARCFOUR olarak da bilinir. RC4 bir akış şifreleme uygulamasıdır. SSL, WEP, WPA gibi birçok güncel uygulamada kullanılmaktadır.

RC4 rasgele olarak ürettiği anahtar akışlarını, hem şifreleme hem de açma işlemi sırasında özel ve ya işlemi ile mesaja uygulamaktadır. Bu projede , RC4 ile metin ve görüntü şifrelenmektedir. Görüntü şifrelenmeden önce 2B Cat Map Kaotik Sistemi ile karıştırılmaktadır. Görüntü karıştırıldıktan sonra RC4 ile şifrelenmektedir. Aynı zamanda bu projede, şifrelenen metin ve görüntüler RC4 kullanılarak geri döndürülebilir. [1]

Anahtar Kelimeler: Cat Map, Görüntü, Kaotik, Metin, RC4, Şifreleme

ABSTRACT

THESIS

TEXT AND IMAGE ENCRYPTION WITH RC4 ALGORITHM

Fırat UCAR

Bilecik Şeyh Edebali University
Engineering Faculty
Department of Computer Engineering

Advisor: Doç. Dr. Cihan KARAKUZU

2018, 42 Pages

Jury	Sign
. . . . .	. . . . .
. . . . .	. . . . .
. . . . .	. . . . .

RC4 was developed in 1987 by Ron Rivest. It is been used in a variety of applications since then. Also known as ARC4 or ARCFOUR. RC4 is a stream encryption application. It is used in many current applications such as SSL, WEP, WPA.

RC4 applies randomly generated key flows to the message either during encryption or decompression by xor operation. In this project, text and image are encrypted with RC4. Before the image is encrypted, mixed with 2B Cat Map Chaotic System.After that, image is encrypted with RC4. At the same time in this project, encrypted texts and images can be recalled using RC4. [1]

Keywords: Cat Map, Chaotic, Encryption, Image, RC4, Text

ÖNSÖZ

Bitirme çalışmamın başından sonuna kadar emeği geçen ve beni bu konuya yönlendiren saygı değer hocalarım ve danışmanım Sayın Doç. Dr. Cihan KARAKUZU'ya ve Arş. Gör. Sefa TUNÇER'e tüm katkılarından ve hiç eksiltmedikleri desteklerinden dolayı teşekkür ederim.

Fırat UÇAR

3 Ocak 2018

İÇİNDEKİLER

ŞEKİLLER TABLOSU

KISALTMALAR

2B : 2 Boyut

CLR : Ortak Dil Çalışması

CRC : Döngüsel Artıklık Kontrolü

DES : Veri Şifreleme Standardı

DLL : Dinamik Bağlantı Kütüphanesi

ICV : Bütünlük Kontrol Değeri

IEEE : Elektrik ve Elektronik Mühendisleri Enstitüsü

KSA : Anahtar Planlama Algoritması

LAN : Yerel Alan Ağı

MIC : Mesaj Bütünlüğü Kodu

MSIL : Microsoft Ara Dili

PRGA : Sahte Rasgele Üretim Algoritması

SSL : Güvenli Oturum Katmanı

TKIP : Geçici Anahtar Bütünlüğü Protokolü

WEP : Kablosuz Denk Mahremiyet

WLAN : Kablosuz Yerel Alan Ağı

WPA : Kablosuz Korumalı Erişim

1 GİRİŞ

Kriptoloji, saklanması veya gönderilmesi gereken mesajların, bilgilerin bir anahtarla belirli bir sisteme göre şifrelenmesi, şifrelenen mesajın anahtarı kullanılarak alıcı tarafından deşifre edilmesidir. Kısaca şifre bilimine kriptoloji denilmektedir. Kriptografi ve kripto analiz olmak üzere iki dala ayrılır. Günümüzde teknoloji çok hızlı geliştiğinden güvenlik sorunlarını da beraberinde getirmektedir. Özel şirketler, askeri kurumlar, devlet kurumları vb. birçok birim arasındaki iletişimin güvenliğini sağlamak için kriptoloji alanındaki gelişmeler büyük önem arz etmektedir. Bu amaçla güvenlik zaafiyetlerini engellemek ve bunu yaparken hızlı iletişimi de sağlamak amacıyla birçok kriptografik yöntem geliştirilmektedir. [2]

Şekil 1.1: Kriptografi

Kriptografik algoritmalar gizlilik, bütünlük, süreklilik, kimlik denetimi, inkar edilememezlik ve izlenebilirlik gibi güvenlik protokollerinin bileşenleri haberleşmede güvenliği sağlamak amacıyla kullanılırlar. Kriptografik sistemler simetrik anahtarlı, asimetrik anahtarlı olmak üzere ikiye ayrılır fakat anahtarsız şifrelemede mevcuttur. Simetrik sistemlerde bir gizli anahtar mevcuttur, bu anahtarın gönderici ve alıcı tarafta bulunması gerekmektedir. Asimetrik sistemlerde gizli bir anahtara ek olarak, açık anahtar mevcuttur. Bu sistemlerde açık anahtar ve gizli anahtarın ikisi ele geçirildiğinde şifre çözme işlemini yapmak mümkündür. Bu sayede, açık anahtarın üçüncü bir şahıs tarafından ele geçirilmesi şifrelenen bilgilerin ele geçirilmesi açısından bir tehdit oluşturmamaktadır.

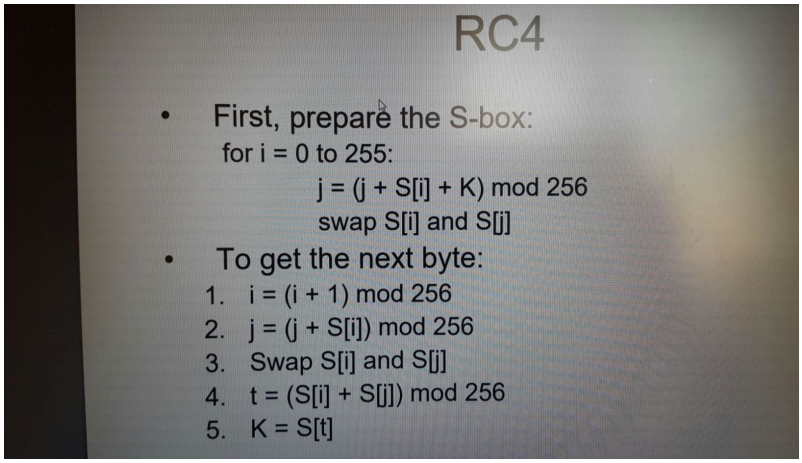
Kaos teorisi sonuçları tahmin edilemeyen sistemleri açıklayan bilimsel bir ilkedir. Genel anlamda 1980'li yıllarda araştırılmıştır. Bir sigara dumanının havada yaptığı şekiller düzensiz ve bağımsız değildir. Sigaranın bu dinamikleri ortamdaki birçok etkene ve parametreye bağlıdır. Ancak bu parametre ve bilgiler çok fazladır ve bunları inceleyip net olarak bir kanıya varmak imkansızdır. Sigara dumanının şeklini bulunan ortamdaki rüzgar, sıcaklık, basınç gibi fiziksel büyüklükler etkileyebilir ve bu faktörlerin birbirine bağlı olabileceği de hesaba katıldığında durum tahmin edilemeyen bir hale gelir.

Kaotik sistemlerin belirli bir frekans aralığında bulunduğu ve bu sınırlar içerisinde hareketlerinin belirlenemez olduğu Lorenz tarafından saptanmıştır. Kaotik sistemleri, belirli bir alana çeken bu yapılara kaotik çekerler adı verilmektedir. Bu çekerlere bakıldığında bu çekerleri meydana getiren eğrilerin, belirli sayıdaki parametreler dizisinin, zaman düzleminde hiçbir zaman iki kez aynı rotayı izlemediği ve bu rotalar arasında kesinlikle küçük de olsa bir farklılık olduğu görülmüştür. Çekerlerin oluşturduğu eğrilerin bu özelliğinden dolayı fraktal yapıda olduğu söylenebilir.

Kaotik sistemler görüntü şifreleme uygulamalarında oldukça yoğun kullanılmaktadır. Bilinen kaotik sistemlerin kullanıldığı ve başarım ve güvenlik bakımından oldukça iyi sonuçlar elde edildiği görülmektedir. Ayrıca birden fazla kaotik sistem kullanılması bu sistemlerin başlangıç değerlerine hassas bağlılığı ve şifrelemede kullanılan parametrelerin fazla olması güvenlik seviyesini oldukça arttırmaktadır. Bu sayede şifrelenen görüntünün istenmeyen şahıslar tarafından ele geçirilmesi, şifrelemede kullanılan parametreleri bilmediği taktirde, neredeyse imkansız hale gelmektedir.

Görüntü şifrelemede kaotik sistemlerin kullanılması, sağladığı bazı üstünlüklerden ve analiz yöntemlerine karşı gösterdiği yüksek dayanıklıktan kaynaklanmaktadır. Bu analizler anahtar güvenliği, histogram analizi, anahtar hassaslığı, görüntüdeki bitişik piksellerin birbiri ile olan ilişki katsayısı, bilginin düzensizlik analizi, şifreleme ve şifre çözme hızıdır. Bu sayede, yapılan her analize karşı doğrudan daha başarılı olduğu söylenemese de, genel anlamda kullanılan geleneksel simetrik ve asimetrik şifreleme algoritmalarına karşı üstünlük sağladığı görülmektedir.

RC4; SSL, WEP, WPA, gibi güncel pekçok uygulamada kullanılan bir akış şifreleme uygulamasıdır. RC4 'ün genel olarak kullanılan algoritması Şekil 1.2'deki gibidir. [1]



Şekil 1.2: RC4 algoritması

Projede metin ve ya görüntü seçimine göre seçilen belge RC4 ile şifrelendi. Bu şifreleme işlemi gerçekleştirilirken seçilen belgenin görüntü olması durumunda önce Cat Map ile karıştırıldı ve ardından RC4 uygulandı. Cat Map ile görüntünün karıştırılması işlemi ileriki bölümlerde detaylı bir şekilde anlatılmıştır.

RC4 ile şifreleme işleme yapılırken bir takım gereksinimler vardır. Bunlardan bir tanesi kullanıcı tarafından bir anahtar girilmesidir. Bu şart şifreleme için olmazsa olmaz bir kuraldır. Çünkü şifreleme işlemleri yapılırken bu girilen anahtar kullanılarak diğer bölümlerde detaylı bir şekilde anlatılacak olan bazı işlemler yapılacaktır. Bu işlemler KSA ve PRGA'dır. Bu iki adımın detaylı anlatımları ve kod kısımları ilerleyen bölümlerde anlatılacaktır.

KSA ve PRGA adımları yapıldıktan sonra seçilen belgenin tipine göre gerekli işlemler yapıldıktan sonra şifreleme işlemi gerçekleştirilecektir.

2 PROGRAMLAMA

Projede görüntünün karıştırılması aşamasında MATLAB , şifrelenmesi aşamasında ise C# programları kullanıldı. Aşağıda MATLAB ve C# programlarının ne oldukları ve kullanım alanları verilmiştir.

2.1 Matlab

MATLAB temel olarak nümerik hesaplama, grafiksel veri gösterimi ve programlamayı içeren teknik ve bilimsel hesaplamalar için yazılmış yüksek performansa sahip bir yazılımdır.

Şekil 2.1: MATLAB

MATLAB adı, MATrix LABoratory (Matrix Laboratuarı) kelimelerinden gelir. MATLAB ilk olarak Fortran Linpack ve Eispack projeleriyle geliştirilen ve bu programlara daha etkin ve kolay erişim sağlamak amacıyla 1970'lerin sonlarında yazılmıştır. İlk başlarda bilim adamlarına problemlerin çözümüne matris temelli teknikleri kullanarak yardımcı olmaktaydı. Bugün ise geliştirilen yerleşik kütüphanesi ve uygulama ve programlama özellikleri ile gerek üniversite ortamlarında(başta matematik ve mühendislik olmak üzere tüm bilim dallarında) gerekse sanayi çevresinde yüksek verimli araştırma, geliştirme ve analiz aracı olarak yaygın bir kullanım alanı bulmuştur. Ayrıca işaret işleme, kontrol, fuzzy, sinir ağları, wavelet analiz gibi bir çok alanda ortaya koyduğu Toolbox adı verilen yardımcı alt programlarla da özelleştirilmiş ve kolaylaştırılmış imkanlar sağlamış ve sağlamaya da devam etmektedir. Web adresi : `http://www.mathworks.com`'dur.

MATLAB komut temelli bir programdır. Command Windows penceresinde » işareti MATLAB'ın komut promptunu gösterir ve bu işaretin bulunduğu satır komut satırı olarak adlandırılır. Bu işaretin hemen yanında yanıp sönen I şeklindeki işaret komut ve metin yazma cursoru yani imlecidir. Bu işaretin olduğu yerde klavyeden giriş yapılabilir demektir. [3]

2.1.1 Matlab kullanım alanları

MATLAB; matematik ve hesaplama işlemlerin, algoritma geliştirme, modelleme, simülasyon ve öntipleme, veri analizi ve görsel efektlerle destekli gösterim, bilimsel ve mühendislik grafikleri, uygulama geliştirme gibi alanlarda kullanılmaktadır.

2.2 C#

Günümüze kadar pek çok programlama dili geliştirilmiştir. Bunlar kullanılacak platformlara göre ya da dil yapısına göre farklı alanlarda kullanılır. Tüm diller arasında özellikle nesnel programlama alanında iki programlama dili insanlık için oldukça önemlidir. Bu dillerin ilki ortak platform olarak çalıştırılabilen Java, ikincisi ise .NET kütüphanesi ile entegre edilerek tüm dillerle ortak platformda programlanabilir ve kolay kodlama yapısı ile C# (C Sharp) programlama dilidir.

Şekil 2.2: C#

C#, yazılım sektörü içerisinde en sık kullanılan iki yazılım dili olan C ve C++ etkileşimi ile türetilmiştir. Ayrıca C#, ortak platformlarda taşınabilir bir programlama dili olan Java ile pek çok açıdan benzerlik taşımaktadır. En büyük özelliği ise .NET Framework platformu için hazırlanmış tamamen nesne yönelimli bir yazılım dilidir. Yani nesneler önceden sınıflar halinde yazılıdır. Programcıya sadece o nesneyi sürüklemek ve sonrasında nesneyi amaca uygun çalıştıracak kod satırlarını yazmak kalır. [4]

Microsoft tarafından geliştirilen C#, C++ ve Visual Basic dillerinde yer alan tutarsızlıkları kaldırmak için geliştirilmiş bir dil olmasına rağmen kısa süre içerisinde nesne yönelimli dillerin içinde en gelişmiş programlama dillerinden biri olmayı başarmıştır. Yazılan program çalıştırıldıktan sonra derleyici tarafından algılanan sınıf ve söz dizimi hataları yazılımcıya ayrı bir ekranda ayrıntısıyla gösterilir ve yazılımcı bu hata penceresinden hataları tespit ederek kolayca düzeltebilir.

2.2.1 .NET Framework nedir?

C# ve .NET Framework bazı kişiler tarafından tek bir kavram olarak algılanmaktadır. Fakat bu iki kavram birbirinden tamamen farklı amaçlar için geliştirilmiştir. C#, nesne yönelimli bir programlama diliyken .NET Framework ise C# için geliştirilmiş bir çalışma ortamıdır. Aslında C# dili, Microsoft tarafından .NET platformu için kod geliştirmek amaçlı tasarlanmış ve C# içerisindeki tüm kütüphaneler .NET platformu içinde tanımlanmış kütüphanelerdir.

.NET platformuda Java diline benzer bir çalışma mantığı izleyerek kodları çalışabilir hale getirmektedir. .NET platformunda kod ilk önce Microsoft Intermediate Language(Microsoft Ara Dili) olarak isimlendirilmiş dosya haline dönüştürülür. Bu dosya içerisinde derlenen kodların Microsoft'un standart haline getirdiği bir assembly dili haline dönüştürür. Bu ara dil de saklanan dosyalar çalıştırılmak istendiğinde ise CLR adı verilen sistem MSIL kodlarını çalıştırır.

2.2.2 C# kullanım alanları

C#; konsol uygulaması geliştirme, windows uygulaması geliştirme, ASP.NET uygulaması geliştirme, web servisleri yazma, mobil uygulama geliştirme ve DLL yazma gibi bir çok alanda kullanılmaktadır.

3 RC4 UYGULAMASI NEDİR ? KULLANIM ALANLARI NELERDİR ?

Bu bölümde projede kullanılan şifreleme uygulaması olan RC4 uygulaması tanıtılmış ve kullanım alanları hakkında bilgi verilmiştir.

3.1 RC4 Nedir ?

Ronald Rivest tarafından geliştirilen RC4 şifreleme algoritması, paylaşılan bir anahtarın güvenli bir şekilde değiştirilmesini gerektiren paylaşılan bir anahtar akışı şifreleme algoritmasıdır. Simetrik anahtar algoritması, şifreleme ve şifre çözme için aynı şekilde kullanılır. Böylelikle veri akışı, üretilen anahtar dizisiyle XOR yapılır. Anahtar dizisine dayanan durum girişlerinin ardışık değişimlerini gerektirdiği için algoritma seri haldedir. Dolayısıyla uygulamalar çok hesaplamayla yoğun olabilir. [1]

3.2 RC4 Nasıl Çalışır ?

RC4 şifreleme algoritmasında anahtar akışı, kullanılan basit metinden tamamen bağımsızdır. Burada girdilerin her biri 0 ile 255 arasındaki sayıların bir permütastonudur. Permütasyon değişken uzunluk anahtarının bir fonksiyonudur. İki sayaç vardır ve her ikisi de algoritmada kullanılan 0'dan başlamıştır.

Algoritma, anahtar kurulum ve şifreleme olmak üzere iki aşamada çalışır. Şekil 3.1'de gösterilen anahtar kurulumu, bu şifreleme algoritmasının ilk ve en zor aşamasıdır. N-bit anahtar kurulumu sırasında (N anahtar uzunluğudur) şifreleme anahtarı, iki diziyi, durum ve anahtarı ve karıştırma işlemlerini kullanarak bir şifreleme değişkeni oluşturmak için kullanılır. Bu şifreleme işlemleri baytların değiştirilmesi, modulo işlemleri ve diğer formüllerden oluşur. Bir modulo işlemi, bölmeden kalanı bulmak için kullanılır.

```
for i ← 0 to 255 do
  S[i] ← i
end
j ← 0
for i ← 0 to 255 do
  j ← j+S[i]+K[i mod len(K)] mod 256
  swap(S, i, j)
end
i ← 0
j ← 0
```

Şekil 3.1: Anahtar oluşturma

Algoritma, 256 baytlık bir durum tablosunu başlatmak için 1 ile 256 bayt arasında değişken uzunluk anahtarı kullanır. Durum tablosu daha sonra Şekil 3.2'deki gibi sahte rasgele bayt üretimi için ve daha sonra şifreli metni vermek üzere düz metinle XOR yapılan bir sahte rasgele akış oluşturmak için kullanılır. Durum tablosundaki her bir öğe en az 1 kez takas edilir.

```
i ← i + 1 mod n
j ← j + S[i] mod n
swap(S, i, j)
return S[ S[i] + S[j] mod n ]
```

Şekil 3.2: Anahtar akışı oluşturma

3.3 RC4 Güçlü Yönleri

RC4 algoritmasının güçlü yönleri aşağıda belirtildiği gibidir.

- Tabloda herhangi bir değerin nerede olduğunu bilmenin zorluğu.
- Tablodaki hangi konumun dizideki her bir değeri seçmek için kullanıldığını bilmenin zorluğu.
- Belirli bir RC4 anahtarının bir kez kullanılması.
- Şifreleme DES'den 10 kat daha hızlı olması.

3.4 RC4 Kullanım Alanları

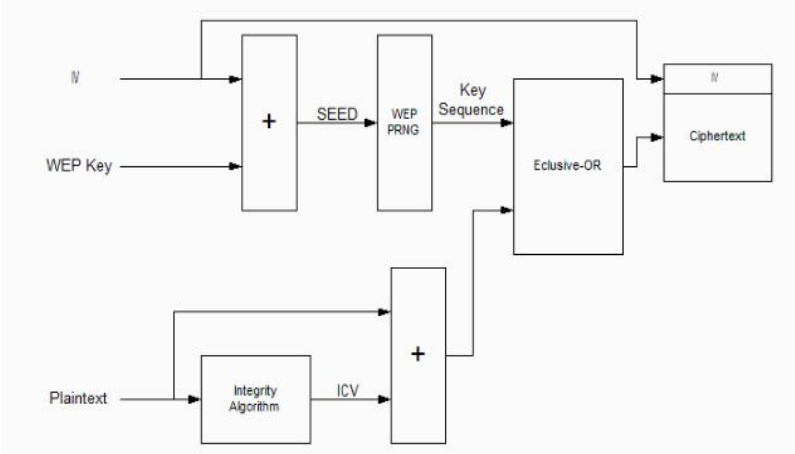
RC4; WEP, WPA gibi güncel bir çok uygulamada kullanılmaktadır.

3.4.1 Wired equivalent privacy

WEP, şifreleme ve kimlik doğrulama için IEEE 802.11 tarafından belirlenmiştir. Standart, WEP'i iki ana bölüm olarak tanımlamıştır. İlki kimlik doğrulama, ikincisi şifreleme kısmıdır. WEP'in hedefleri şunlardır :

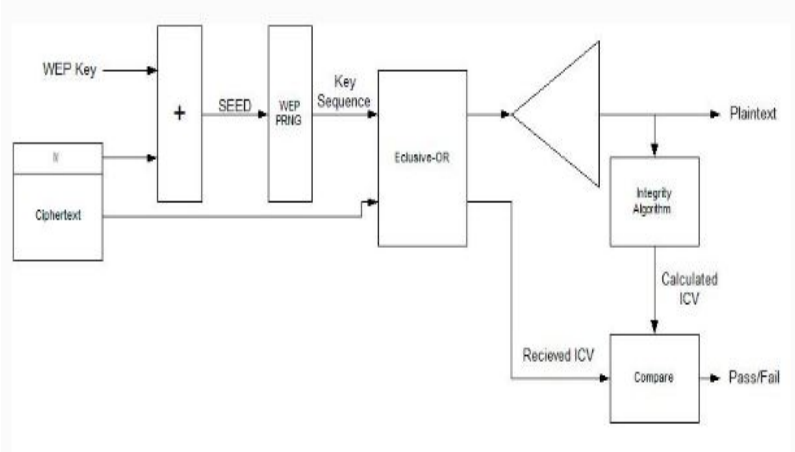
- Doğru WEP anahtarı bulunmadığı için yetkili kullanıcıların erişmesini engelleyerek ulaşılan erişim kontrolü
- Gizlilik, WLAN veri akışını şifrelemek için WEP anahtarı kullanılarak elde edilir ve yalnızca doğru WEP anahtarına sahip olanlar şifresini çözebilir

WEP tarafından kullanılan şifreleme işlemi Rivest Cipher 4 (RC4)'tür. Saldırganlıktan ve ya yetkisiz veri değişikliğinden korumak için kullanılan ICV oluşturmak için düz metin üzerinde CRC-32 kullanan bir bütünlük algoritmasıda vardır. Şekil 3.3'de WEP'in şifreleme algoritması gösterilmiştir. [5]



Şekil 3.3: WEP şifreleme algoritması

Şekil 3.4'de ise WEP şifrelemesinin geri dönüşüm algoritması verilmiştir.



Şekil 3.4: WEP şifreleme geri dönüşüm algoritması

3.4.2 Wireless protected access

IEEE 802.11'i kablosuz ağ standardı, LAN'ı güvenliğindeki gelişmeleri belirtir. Yeni IEEE 802.11'i standardı onaylanırken, kablosuz ürün satıcıları WPA olarak bilinen, çeşitli sistemlerin birlikte çalışmasına olanak veren geçici bir standart üzerinde anlaşmıştır. Geniş şekilde kullanılan bu iki tip WPA standardı WEP'in zayıf yönlerini kapatmak için geçici olarak oluşturulmuştur. Mevcut cihazlar güncellenirse bu protokolü kullanabilir. Günümüzdeki cihazlarda desteği eklenmiş durumdadır. WPA'nın WEP'e tercih edilmesinde üç önemli sebep vardır. Bunlar;

- 802.1X/EAP tabanlı karşılıklı asıllama sağlamaktadır.
- WEP'e göre daha güçlü bir şifreleme yöntemi olan TKIP'i desteklemektedir.
- Veri bütünlüğü için MIC yöntemini kullanmaktadır.

TKIP, çokça tatbik edilen yeni şifreleme protokolüdür. TKIP'in geliştirilmesindeki en büyük etken, WEP tabanlı donanımının güvenliğinin arttırılması ve güncellenmesidir. Genel olarak, WEP kullanan donanımların yonga setleri RC4 şifreleme için donanım desteği sağladı. Donanıma yoğun uygulanan şifreleme ile yazılım donanım ve firmware güncellemeleri geri kalanını mümkün kılmıştır. TKIP, WEP'in temel yapısına ve işlemlerine sahiptir. WEP tabanlı çözümlere karşılık bir yazılım güncellemesi olarak tasarlanmıştır. Esas olarak WEP kusurlu olarak gösterildiği için protokol onu WEP'ten ayırabilmek için yeniden adlandırmıştır. TKIP yukarıda belirtildiği gibi RC4 akış şifrelemesini de kullanır. Sebebi WPA tam bir güvenlik standardı olarak gelişmemiştir. [6]

4 2B CAT MAP KAOTİK SİSTEMİ

Projede görüntü şifreleme işlemi yapılırken, şifrelemeden önce seçilen görüntü MATLAB'da 2B Cat Map kullanılarak karıştırılmıştır. Karıştırılan görüntü daha sonra şifrelenmek üzere programda belli işlemlere tabii tutulmuştur.

4.1 2B Cat Map Nedir ? Nasıl Çalışır ?

Görüntü karıştırma permütasyon evresinde genellikle iki boyutlu üç tip kaotik harita kullanılmaktadır. Bunlar Standart Map, Cat Map ve Genelleştirilmiş Baker Map haritalarıdır. Cat Map literatürde en yaygın kullanılan haritadır. M x N boyutlu bir görüntü ve bu görüntüye ait piksel değerlerinin koordinatları C=(x,y)| x,y =1,2..,N olarak belirlenirse Cat Map aşağıdaki gibi tanımlanır. [2]

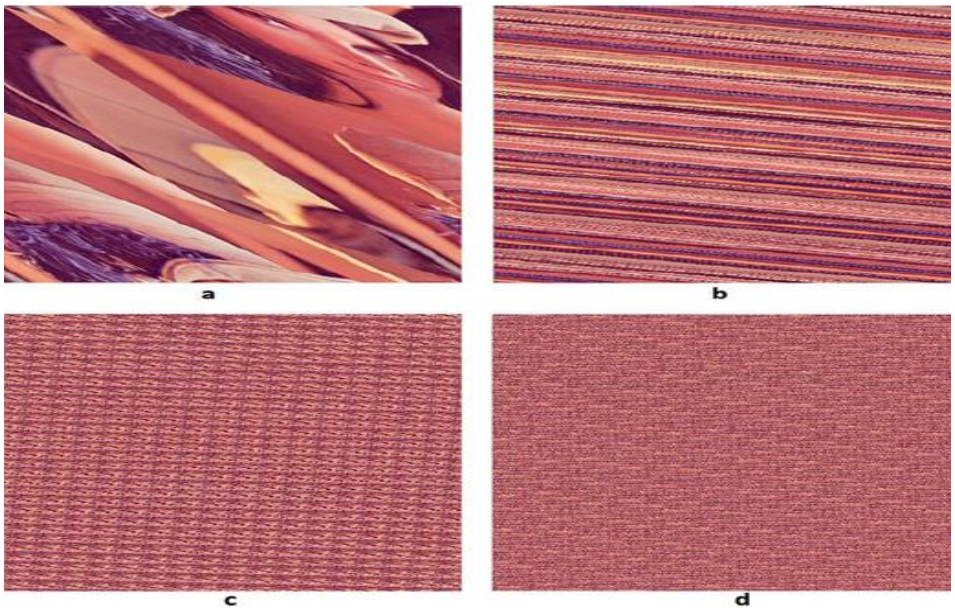
$$\begin{pmatrix} x' \\ y' \end{pmatrix} = Q \begin{pmatrix} x \\ y \end{pmatrix} mod(N) = \begin{bmatrix} 1 & p \\ q & pq+1 \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} mod(N)$$

Şekil 4.1: Cat map görüntü karıştırma algoritması

Şekil 4.1'de gösterilen denklemde Cat Map kontrol parametreleri olan p ve q pozitif tamsayılardır. (x,y) ve (x',y') değerleri ise sırasıyla koordinat değerlerinin orjinal ve yeni pozisyonlarıdır. Burada det(Q)=1 olduğundan alan korunur, yani herhangi bir koordinat birbiriyle çakışmaz ve herhangi bir kayıp meydana gelmez.

Şekil 4.2: Orjinal Lena görüntüsü

2B Cat Map, sınırları belirli bir alanda sürekli farklı koordinat değerleri üreterek görüntüdeki piksellerin yerlerinin değişimini sağlamaktadır. Yukarıdaki şekildeki Lena görüntüsü 2B Cat Map sistemine giriş olarak uygulandığında Şekil 4.3'deki gibi farklı şekillerde görüntü pikselleri karıştırılabilmektedir. Görüntü piksellerinin karışımındaki farklılık 2B Cat Map sisteminde bulunan p ve q parametlerinden kaynaklanmaktadır. Şekil 4.3'de elde edilen görüntülerde p=1 ve q=1 seçildiğinde (a) görüntüsü, p=5 ve q=7 seçildiğinde (b) görüntüsü, p=22 ve q=30 seçildiğinde (c) görüntüsü, p=401 ve q=401 seçildiğinde (d) görüntüsü elde edilmektedir. Bu değerlerin değişiminden de anlaşılacağı üzere p ve q değerleri arttırıldıkça görüntüdeki pikseller daha homojen karışmaktadır.



Şekil 4.3: Karıştırılmış Lena görüntüleri

Piksellerin koordinatlarının değişmesi, görüntü şifreleme işlemlerinde büyük kolaylık sağlamaktadır. Orjinal görüntüde bulunan bitişik piksel değerleri birbirine çok yakın olacağından, bu kısımlar şifrelendiğinde birbirine yakın değerler oluşturabilmektedir. Fakat 2B Cat Map uygulanan görüntüde bitişik piksel değerleri farklılaşacaktır. Bu sayede daha sağlam bir şifreleme yapılabilecektir. Görüntü karıştırma işlemleri tamamlandıktan sonra görüntünün tekrar eski haline getirilmesi Şekil 4.4'deki gibi tanımlanır.

$$\begin{bmatrix} x_n \\ Y_n \end{bmatrix} = \begin{bmatrix} 1+ab & -a \\ -b & 1 \end{bmatrix} \begin{bmatrix} x_{n+1} \\ Y_{n+1} \end{bmatrix} \bmod 1$$

Şekil 4.4: 2B Cat Map ile karıştırılan görüntünün geri dönüştürülmesi

5 ŞİFRELEME İŞLEMLERİ

Bu bölümde C# programı ile RC4 şifrelemesi yapılırken programın sadece şifreleme aşamaları ve kodları anlatılmıştır.

5.1 RC4 Sınıfının Oluşturulması

Şifreleme işlemlerine başlamadan önce C# 'da RC4 adı verdiğimiz bir sınıf oluşturuyoruz. Bu sınıfta RC4 algoritmasının kullanılacağı şifreleme işlemleri bulunmaktadır. RC4 sınıfında kullanılmak üzere dışarıdan anahtar ve şifrelenecek (ve ya geri döndürülecek) metin alınmaktadır. İlk olarak sınıfımızda dışarıdan alınan bu iki değer set ve get edilir. Daha önce anlatıldığı gibi RC4 algoritması iki farklı aşamadan oluşmaktadır. Bu sınıfımız içinde ilk olarak sbox oluşumundan bahsedeceğiz. Sbox bizim bu sınıfta oluşturduğumuz bir dizi olup, şifreleme işlemleri gerçekleşirken bu dizi baz alınacaktır. Sbox girilen anahtara göre Şekil 5.1'deki gibi oluşturulmaktadır.

```
for i ← 0 to 255 do
  S[i] ← i
end
j ← 0
for i ← 0 to 255 do
  j ← j+S[i]+K[i mod len(K)] mod 256
  swap(S, i, j)
end
i ← 0
j ← 0
```

Şekil 5.1: Sbox oluşumu

Şekil 5.1'de görülen algoritmayı açıklayacak olursak; algoritma gereği döngü 256'ya kadar gitmektedir. Bunun için N değişkeni, RC4 sınıfının başında tanımlanmış olup, tüm program boyunca algoritmanın gereği olarak sabit değer 256 tanımlanmıştır. Sbox oluşumu için kullanıcının girdiği anahtar kelime kullanılarak yeni bir dizi oluşturulmuştur. For döngüsü ile anahtarın harflarinin int tipindeki değerleri, anahtar kelimenin uzunlu-

ğuyla mod alınarak 256 boyutlu bir yeni dizi oluşturmaktadır. Aynı for döngüsü içinde sbox'da oluşturulmaktadır. Diğer for döngüsünde yine 256'ya kadar algoritma gereği ilk olarak 0 olarak tanımlanan b değeri, sbox değeri ve yeni dizinin değerleri toplanarak değiştirme işlemleri uygulanır. Böylelikle ilk olarak 1'den başlayıp 256'ya kadar değer alan sbox yeni değerlerini almış olur. Böylelikle iki aşamadan meydana gelen RC4'ün ilk aşaması tamamlanmış olur.

RC4'ün algoritmasında bulunan ikinci aşama ise Sifrele() metodu ile tanımlanmıştır. ilk aşamada üretilen sbox Sifrele() metodunun en başında çağırılır ve çeşitli işlemler ile Şekil 5.2'deki algoritmadaki gibi kullanılır.

$$
\begin{aligned}
&\mathrm{i} \leftarrow \mathrm{i} + 1 \bmod \mathrm{n} \\
&\mathrm{j} \leftarrow \mathrm{j} + \mathrm{S[i]} \bmod \mathrm{n} \\
&\mathrm{swap(S,\, i,\ j)} \\
&\mathrm{return\ S[\ S[i]\ + S[j]\ \bmod n\ ]}
\end{aligned}
$$

Şekil 5.2: Şifreleme aşaması

Şekil 5.2'deki algorima Sifrele() metodu ile birlikte kullanılırken işlem kolaylığı ve performans açısından StringBuilder sınıfı kullanılmıştır. RC4 algoritmasının genel yapısında bulunan gerekliliklerden ötürü yukarıda da görüldüğü gibi bazı mod alma işlemleri ve yer değiştirme işlemleri uygulanmıştır. En sonda oluşturulan değer ile şifrelenecek ve ya geri döndürelecek metod XOR işlemine girer. Böylelikle şifrelenmeden önceki son aşama olan bu aşamada dönüştürülmeden önce şifrelenmiş değerler belirlenmiş olur.

RC4 sınıfında StrToHexStr ve HexStrToStr olmak üzere iki metod daha bulunmaktadır. Fakat bu metodlar form ekranındaki butonlara tıklandıkça kullanılacağı için daha sonra bahsedilecektir.

5.2 Form Ekranının Tasarlanması

Bu bölümde form ekranında bulunan Toolbox'ların neler olduğu, ne amaçla form ekranında bulundukları ve form ekranındaki yerleşimleri hakkında bilgi verilecektir. Öncelikle, form ekranında bulunan Toolbox'lar aşağıdaki gibidir:

- Buton
- OpenFileDialog
- SaveFileDialog
- Label
- PictureBox
- TextBox

5.2.1 Buton kullanımı

Form ekranında birden fazla buton kullanılmıştır. Kullanılan bu butonlar şifrelenecek belgeyi seçmede, seçilen belgeyi şifrelemede, şifrelenen belgeyi geri seçmede, seçilen belgeyi geri döndürmede ve kayıt işlemlerinde kullanılmaktadır.

5.2.2 OpenFileDialog kullanımı

OpenFileDialog kullanımı şifrelenecek belgeyi seçmek için pencere açılmasında ve kayıt işlemlerinde kayıt yerinin belirlenmesi için pencere açılmasında kullanılmaktadır.

5.2.3 SaveFileDialog kullanımı

SaveFileDialog adının Türkçe karşılığındanda anlaşılacağı gibi kayıt işlemlerinin yapılması için gereklidir. Şifrelenen metni belge haline dönüştürüp kaydetme işleminde ve şifrelenmiş metinin geri döndürülüp asıl haline ulaştıktan sonra kayıt etme işlevlerinde kullanılmaktadır.

5.2.4 Label kullanımı

Form ekranında gerekli görülen yerlere başlık eklemek için kullanılmaktadır.

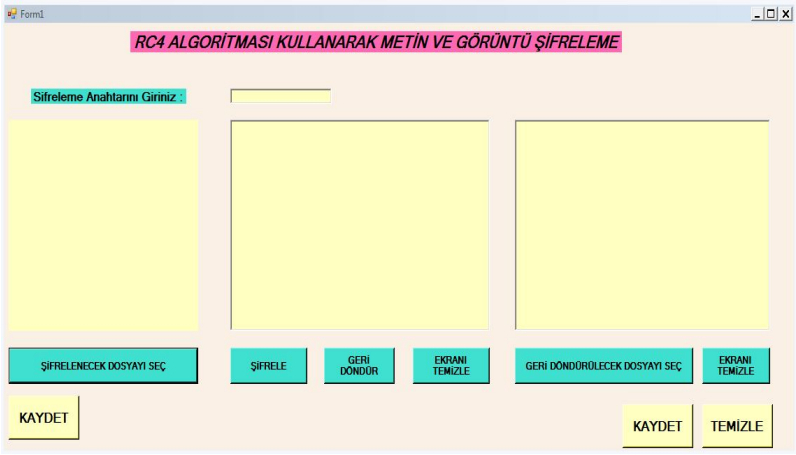
5.2.5 PictureBox kullanımı

Form ekranında şifrelenecek belgeyi seçerken, seçilen belgenin bir görüntü olması halinde pictureBox ekranında görüntülenmesi için kullanılmaktadır.

5.2.6 TextBox kullanımı

Form ekranında kullanıcı tarafından girilecek olan anahtar kelimeyi yazmada, gerek belge olarak, gerekse kullanıcı tarafından şifrelenecek bir metin girilmesi için ve şifrelenmiş metnin görüntülenmesi için kullanılmaktadır.

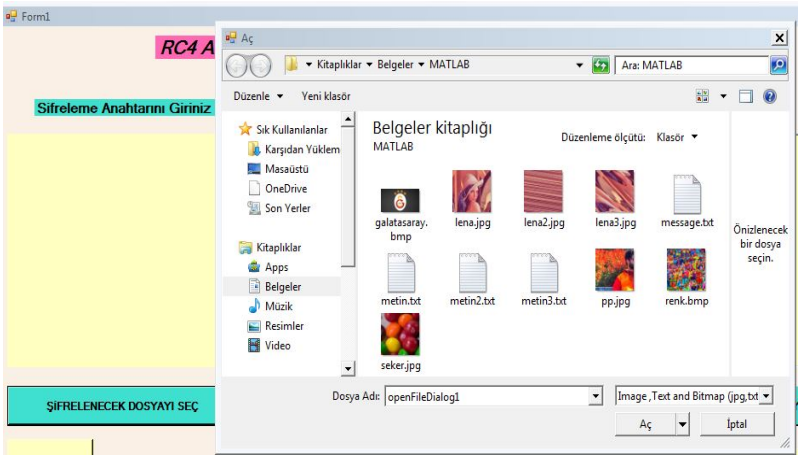
Tüm bu Toolbox'ların kullanımı sonucunda form ekranı Şekil 5.3'deki gibi meydana gelmiştir.



Şekil 5.3: Form ekranı

5.3 Şifreleme İşlemi

RC4 sınıfının ve form ekranının oluşumundan sonra şifreleme için gerekli kodların yazılmasında bir engel kalmamıştır. İlk olarak şifreleme yapabilmek için eğer klavyeden bir metin girilerek şifreleme yapılmak istenmiyorsa şifreleme yapabilmek için bir dosya seçilmelidir. Bu dosyayı seçmek için form ekranında bir buton oluşturulmuştur. ŞİFRELENECEK DOSYAYI SEÇ butonu ile şifrelenmesi istenen dosya Şekil 5.4'de gösterildiği gibi seçilmektedir.



Şekil 5.4: Şifrelenecek dosyanın seçilmesi

Bu seçim işlemi OpenFileDialog kullanılarak gerçekleştirilmektedir. Dosya seçilirken metin ve görüntü şifrelemesi yapılacağından ötürü çıkan sonuçlar filtrelenmiştir. Sadece .jpg, .bmp ve .txt dosyaları görüntülenecektir. Try catch yapısı kullanılarak seçilen dosyanın metin ve ya görüntü olmasına göre ayrı ayrı yapılacak işlemler belirlenmiştir. Eğer seçilen belge bir görüntü ise bu görüntünün şifrelenebilmesi için byte tipine dönüştürülmesi gerekmektedir. Bunun için seçilen görüntü ImageToBase64 metoduna gönderilerek görüntünün byte tipine çevrilmesi sağlanır. Ardından byte tipine çevrildikten sonra değerler şifrelenmek için şifreleme textBox'ına yazdırılır. Görüntünün şifrelenmeden önce gönderildiği ImageToBase64 metodu Şekil 5.5'deki gibi verilmiştir. [7]

```
using (MemoryStream ms = new MemoryStream())

    image.Save(ms, format);
    byte[] imageBytes = ms.ToArray();

    string base64String = Convert.ToBase64String(imageBytes);
    return base64String;
```

Şekil 5.5: ImageToBase64 metodu

MemoryStream sınıfı kullanılarak yapılan bu dönüşüm sonucunda string tipinde çıkan sonuç şifrelemenin yapılacağı textBox'a yazdırılır. Bu aşamadan sonra şifreleme yapabilmek için ŞİFRELE butonu devreye girer.

Form ekranında bulunan ŞİFRELE butonuna basarak şifreleme işlemi başlar. Butona tıkladıktan sonra Şekil 5.6'de şifreleme ekranındaki metin şifrelenerek şifrelenmiş alana yazdırılır.

Şekil 5.6: Şifreleme işlemi

ŞİFRELE butonunun içinde bulundurduğu kodda öncelikle kullanıcı tarafından girilmesi zorunlu olan anahtarın girilip girilmediği kontrol edilir. Eğer bu alan boş bırakılmış ise Message.Show ile bir hata mesajı gösterilir ve şifreleme anahtarının boş geçilemeyeceği bildirilir. Eğer şifreleme anahtarı olması gerektiği gibi girilmiş ise öncelikle RC4 sınıfından bir nesne oluşturulur ve bu sınıfa girilen şifreleme anahtarı ve şifrelenmek için oluşturulan TextBox'daki metin gönderilir. RC4 sınıfında meydana gelen işlemler sonucunda çıkan sonuç Hex tipine çevrilmek üzere StrToHexStr metoduna gönderilir ve çevrilen değer şifreli metinin gösterilmesi için oluşturulan TextBox'a yansıtılır.

Şifrelenmek üzere gönderilen metin farklı tiplerde yansıtılabilir. Fakat yapılan araştırma sonucu en yaygın kullanımı Hex tipi olduğundan burada şifreli metin Hex tipinde gösterilmiştir. Hex tipine çevirme işlemi Şekil 5.7'de gösterildiği gibidir.

```
public static string StrToHexStr(string str)
{
    StringBuilder sb = new StringBuilder();
    for (int i = 0; i < str.Length; i++)
    {
        int v = Convert.ToInt32(str[i]);
        sb.Append(string.Format("{0:X2}", v));
    }
    return sb.ToString();
}
```

Şekil 5.7: Şifrelenen metinin stringden hex tipine dönüşümü

Bu dönüşüm işlemi yapılırken yine kolaylık ve performans açısından StringBuildir sınıfı kullanılmıştır. Burada gönderilen metin int tipine çevrildikten sonra StringBuilder sınıfına ait string.Format metodu ile hex tipinde gösterilmiştir. string.Format metodu farklı bir tipteki değişkeni başka bir tipte göstermeye yarar. Nitekim burada int tipindeki her bir değeri 2 haneli hex değerler olarak gösterilmiştir. [8]

Böylelikle şifreleme işlemi tamamlanmış olur. En başından beri konuyu ele alıp özetleyecek olursak, şifrelenenmek üzere belge seçildikten sonra seçilen belge görüntü ise önce byte tipine çevrildi ve daha sonra şifrelenmek üzere textBox'a yazdırıldı. Aynı zamanda şifreleme için şifreleme anahtarının girilmesi sağlandı. Eğer seçilen dosya metin belgesi ise seçilen metin belgesi textBox'a yazdırıldı. TextBox'daki bu değerler hex tipine çevrilerek şifrelenmiş alanda bulunan textBox'a yazdırıldı ve şifreleme işlemi tamamlanmış oldu.

5.4 Şifreli Metinin Kaydedilmesi

Şifreleme işlemleri tamamlandıktan sonra form ekranında bulunan iki KAYDET butonundan sağ alttaki ile şifreli metin kaydedilebilmektedir. Kayıt edilen bu şifreli metin daha sonra geri döndürülmek üzere kullanılacaktır.

6 GERİ DÖNDÜRME İŞLEMLERİ

Bu bölümde RC4 ile şifrelenen şifreli metni tekrar eski haline döndürme işlemlerinden bahsedilecektir.

6.1 Kaydedilen Şifreli Metinin Seçilmesi

Şifreleme işlemleri tamamlandıktan sonra şifreli metin KAYDET butonu ile kaydedildi. Geri döndürme işlemi yapılırken form ekranında bulunan GERİ DÖNDÜRÜLECEK DOSYAYI SEÇ butonu ile şifrelenmiş metin OpenFileDialog ile seçilerek şifrelenmiş metin form ekranına yansıtılır. Bu yansıtma işlemi StreamReader sınıfı ile Şekil 6.1'de gösterildiği gibi gerçekleşmektedir.

```
string dosya_adı = openFileDialog2.FileName;

System.IO.StreamReader metin = new System.IO.StreamReader(dosya_adı);

textBox3.Text = metin.ReadLine();

metin.Close();
```

Şekil 6.1: StreamReader sınıfının kullanımı

Şekil 6.1'de görülen StreamReader sınıfı OpenFileDialog ile seçilen metni okur ve form ekranında belirtilen yere yazar. Bu aşamadan sonra geri döndürülmesi istenen form ekranına gelmiştir. Artık GERİ DÖNDÜR butonu ile şifreli metin ilk haline dönüştürülecektir.

6.2 Geri Döndürme İşlemi

Şifrelenen metin geri döndürülmek üzere seçildikten sonra geri döndürme işleminin yapılabilmesi için girilen anahtar kelimenin şifreleme yapılırken kullanılan anahtar kelime

ile eşdeğer olması gerekmektedir. Aksi taktirde şifreli metin yeni anahtar kelimeye göre döndürülecektir ve bu da olumsuz sonuç alınmasına neden olacaktır.

Geri döndürme işlemi yapılırken şifreli metin hex tipinde bir metindi. Öncelikle hex tipindeki metinin stringe çevrilmesi gerekmektedir. Bunun için HexStrToStr metodu Şekil 6.2'de gösterildiği gibi kullanılmıştır. [9]

```
StringBuilder sb = new StringBuilder();
for (int i = 0; i < hexStr.Length; i += 2)
{
    int n = Convert.ToInt32(hexStr.Substring(i, 2), 16);
    sb.Append(Convert.ToChar(n));
}
return sb.ToString();
```

Şekil 6.2: HexStrToStr metodunun kullanımı

Şekil 6.2'de gösterilen metod sayesinde hex tipinde şifrelenmiş olan şifreli metin string tipine dönüştürülür. Böylelikle geri döndürme işlemi için son bir aşama kalmış olur. String türüne çevrilen metin RC4 sınıfına girilen anahtar kelime ile tekrar gönderilir ve işlemler sonucunda XOR'lanarak eski haline dönüştürülmüş olur. Dönüştürülen bu düz metin try catch ile iki farklı durum için incelenmektedir. Eğer şifreleme yapılan dosya bir metin ise geri döndürme işlemleri sonucunda o metin form ekranında belirtilen yere yazdırılacaktır. Ama eğer şifrelenmesi için seçilen dosya bir görüntü ise geri döndürülen metin Base64ToImage metodu ile tekrar bir görüntü haline getirilir ve form ekranında bulunan pictureBox'ta gösterilir. Böylelikle seçilmesi muhtemel iki durum içinde geri döndürme işlemleri tamamlanmış olur.

Geri döndürme işlemleri tamamlandıktan sonra form ekranında gösterilen daha önce 2B Cat Map ile karıştırılmış görüntü, form ekranının sol alt kısmında bulunan KAYDET butonu ile tekrar karıştırılıp eski haline getirilmesi için kaydedilmektedir.

7 SONUÇLAR VE ÖNERİLER

Sonuç olarak bu projede RC4 algoritması kullanılarak metin ve görüntü şifrelemesi C# programlama dili yardımı ile şifrelenmiş ve daha sonra geri döndürülmüştür. Görüntü şifrelemesi yapmadan önce ise MATLAB yardımı ile 2B Cat Map Kaotik Sistemi kullanılarak görüntü karıştırılmış ve şifrelenme işleminden sonra geri döndürme işlemide uygulandıktan sonra tekrar Cat Map ile karıştırılan görüntü eski haline dönüştürülmüştür. Şekil 7.1'de programın son hali gösterilmiştir. Burada seçilen görüntü daha anlaşılır olması açısından Cat Map ile karıştırılmadan şifrelenmiştir.

Şekil 7.1: Lena görüntüsünün RC4 ile şifrelenmesi

RC4 algoritmasında şifrelenecek dosyalar çeşitli Kaotik Sistemler kullanılarak şifreleme işlemi çözülmesi daha güç bir hale getirilebilir. **??**

8 EKLER

Bu proje sürecinde, proje konusuna dahil olmayan Runge Kutta yönteminin MATLAB ortamında uygulaması yapılmıştır. Runge Kutta yöntemi EK-1'de, rapor içerisinde konusu ve uygulaması geçen fakat içeriğindeki kod kısmı hakkında çok fazla bilgi verilmeyen MATLAB ortamında Cat Map ile görüntü karıştırma uygulamasının kodu EK-2'de verilmiştir.

8.1 EK-1

Sayısal analizde Runge Kutta yöntemleri, adi diferansiyel denklemlerin çözüm yaklaşımları için açık yinelemeli yöntemler ailesinin önemli bir tipidir. Bu yöntem 1900'lü yıllarda C. Runge ve M. W. Kutta adlı matematikçiler tarafından geliştirilmiştir. 4. dereceden klasik Runge Kutta yöntemi RK4 veya Runge-Kutta yöntemi olarak adlandırılır. Bu yöntem sıkça kullanılır. 4. dereceden Runge Kutta yöntemi aşağıdaki Şekil 8.1'de gösterildiği gibi tanımlanır.

$$
\begin{aligned}
k_1 &= f(t_n, y_n) \\
k_2 &= f(t_n + \frac{h}{2}, y_n + \frac{h}{2} \cdot k_1) \\
k_3 &= f(t_n + \frac{h}{2}, y_n + \frac{h}{2} \cdot k_2) \\
k_4 &= f(t_n + h, y_n + h \cdot k_3) \\
y_{n+1} &= y_n + \frac{h}{6} \cdot (k_1 + 2 \cdot k_2 + 2 \cdot k_3 + k_4)
\end{aligned}
$$

Şekil 8.1: Runge Kutta yöntemi

Böylece bir sonraki değer, o anki değerine ve aralığın büyüklüğüyle tahmini eğimin çarpımının eklenmesiyle elde edilir. Bu eğim, eğimlerin ağırlıklı ortalamasıdır.

k1 aralığın başlangıcındaki eğimdir.

k2 aralığın orta noktasındaki eğimdir.

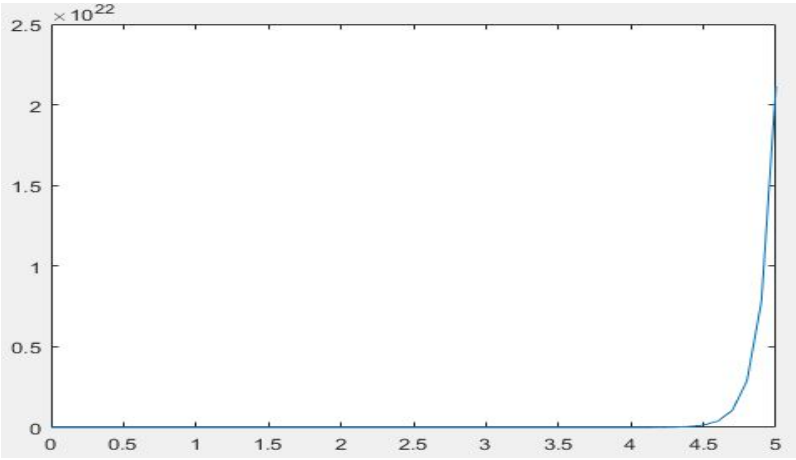
k3 yine orta noktadaki eğimdir. Ancak **k2** eğiminden elde edilir.

k4 aralığın sonundaki eğimdir ve **k3** eğiminden yararlanılarak bulunur.

Runge Kutta yönteminin MATLAB kodu Şekil 8.2'de, ekran çıktısı ise Şekil 8.3'de gösterilmiştir.

```
function [  ] = runge_kutta()
a=10.0;
h=0.1;
xfinal=5;
x(1)=0;
y(1)=5;

f=@(x,y)  a*(y-x);

for i=1:ceil(xfinal/h)
    x(i+1)=x(i)+h;

    k1=f(x(i)       ,y(i)          );
    k2=f(x(i)+0.5*h,y(i)+0.5*k1*h);
    k3=f(x(i)+0.5*h,y(i)+0.5*k2*h);
    k4=f(x(i)+    h,y(i)+    k3*h);

   y(i+1)=y(i)+h/6*(k1 + 2*k2 + 2*k3 + k4);

end
plot(x,y)

end
```

Şekil 8.2: Runge Kutta yöntemi MATLAB kodu



Şekil 8.3: Runge Kutta yöntemi ekran çıktısı

8.2 EK-2

MATLAB ile 2B Cat Map görünrü karıştırma ve görüntüyü geri döndürme kodları aşağıda Şekil 8.4 ve Şekil 8.5'de verilmiştir.

```
function [] = cat_map_rgb(p,q)
im=imread('lena.jpg');

 N=size(im,1);

  X=zeros(size(im));


for i=1:N
    for j=1:N
          newi=mod(((i-1)+ p*(j-1)),N)+1;
          newj=mod((q*(i-1)+ (p*q+1)*(j-1)),N)+1;

             X(newi,newj,1)=im(i,j,1);
             X(newi,newj,2)=im(i,j,2);
             X(newi,newj,3)=im(i,j,3);

    end
end
X=uint8(X);
imtool(X);

end
```

Şekil 8.4: 2B Cat Map ile görüntü karıştırma

```
function [  ] = cat_map_reverse( p,q )
im=imread('lena3.jpg');
 N=size(im,1);
 X=zeros(size(im));


for i=1:N
    for j=1:N
         newi=mod(((p*q+1)*(i-1) + (-p)*(j-1)),N)+1;
        newj=mod(((-q)*(i-1) + (j-1)),N)+1;

            X(newi,newj,1)=im(i,j,1);
            X(newi,newj,2)=im(i,j,2);
            X(newi,newj,3)=im(i,j,3);

    end
end
 X=uint8(X);
imtool(X);

end
```

Şekil 8.5: 2B Cat Map ile karıştırılan görüntünün reverse edilmesi

KAYNAKLAR

[1] Wikipedia, https://en.wikipedia.org/wiki/RC4

[2] Ulusal Tez Merkezi, Arş. Gör. Sefa TUNÇER Kaotik Sistemler Tezi

[3] http://www.teknokolikler.com/2011/11/
c-nedir-c-temelleri-nelerdir.html

[4] http://www.teknokoliker.com/2011/11/
c-nedir-c-temelleri-nelerdir.html

[5] https://www.vocal.com/secure-communication/
wired-equivalent-privacy-wep/

[6] http://www.emo.org.tr/ekler/7600d163fa81512_ek.pdf

[7] https://stackoverflow.com/questions/21325661/
convert-image-path-to-base64-string

[8] Microsoft, https://social.msdn.microsoft.com/Forums/
en-US/74fdc1b9-9074-4c49-b90d-fbd1947c2e00/
string-to-hexadecimal?forum=Vsexpressvcs

[9] Microsoft, https://msdn.microsoft.com/tr-tr/library/
aka44szs(v=vs.110).aspx

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Fırat UÇAR

Uyruğu : TC

Doğum Yeri ve Tarihi: Sultanbeyli 10.08.1994

Adres : Hamidiye mah. Balcı sok. No:6 Çekmeköy/İstanbul

Telefon : 05077687574

e-mail : firatucar94@gmail.com

EĞİTİM DURUMU

Lisans Öğrenimi : BŞEÜ Bilgisayar Mühendisliği Bölümü

Bitirme Yılı : 2017-2018

Lise : Altınay Anadolu Lisesi

İŞ DENEYİMLERİ

Yıl : 2015-2016

Kurum : Mutfark Teras Cafe - Optimal Ormancılık

Stajlar : KOÇ Sistem - Bilecik Şeyh Edebali Üniversitesi Bilgisayar Mühendisliği

YABANCI DİLLER: : Spor, müzik

YABANCI DİLLER: : İngilizce