T.C.

SELÇUK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

ŞİFRELEMEDE KAOTİK SİSTEMİN KULLANILMASI

OĞUZ FINDIK

YÜKSEK LİSANS TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİMDALI

Konya - 2004

T.C.

SELÇUK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

ŞİFRELEMEDE KAOTİK SİSTEMİN KULLANILMASI

OĞUZ FINDIK

YÜKSEK LİSANS TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ
ANA BİLİM DALI

KONYA, 2004

T.C.

SELÇUK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

ŞİFRELEMEDE KAOTİK
SİSTEMİN KULLANILMASI

OĞUZ FINDIK

YÜKSEK LİSANS TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİMDALI

Bu tez 22.07.2004 tarihinde aşağıdaki jüri tarafından oybirliği / oyçokluğu ile kabul edilmiştir

Prof.Dr.Ferruh YILDIZ	Doç.Dr.Ahmet ARSLAN	Doç.Dr.Şirzat KAHRAMANLI
(Üye)	(Üye)	(Üye)

ÖZET

Yüksek Lisans Tezi

ŞİFRELEMEDE KAOTİK
SİSTEMİN KULLANILMASI

OĞUZ FINDIK

Selçuk Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Ana Bilim Dalı

Danışman: ***Doç. Dr. Şirzat KAHRAMANLI***

2004, 86 Sayfa

Jüri:

Prof. Dr. Ferruh YILDIZ
Doç. Dr. Şirzat KAHRAMANLI
Doç. Dr. Ahmet ARSLAN

Son yıllarda internet kullanımının yaygınlaşması bir takım güvenlik sorunlarını da beraberinde getirmiştir. Bunun başlıca sebebi, İnternet'in açık bir sistem olması ve üzerinde dolaşan verinin istenmeyen kişiler tarafından elde edilmesine uygun olmasıdır.

İnternet'te yollanan veri paketleri birçok halka açık ağdan geçer, bu da bu paketlere ulaşmayı mümkün kılar. Bu durum son derece gizli bilgiler internet 'te iletilirken, önemli bir kaygı halini alır. Bu tür bilgileri korumak mümkün olmadıkça, internet; iş yapmak veya kişisel yazışmalarda bulunmak için asla güvenli bir yer olmayacaktır.

Şifreleme bilgisayar ağlarında haberleşme ve dosya güvenliğini sağlamak amacıyla kullanılan bir araçtır. Bilgisayar kullanan insanlar çoğaldıkça kişisel veya özel bilgilere erişmek isteyen kişiler çoğalmaktadır. Bu sebepten dolayı şifrelemenin önemi de gün geçtikçe artmaktadır.

Bu çalışmada güvenli ve hızlı bir şifreleme gerçekleştirebilmek için Kaotik sistemler kullanılmıştır. Kaotik harita oluşturmak için E.Lorenz'in hava tahminleri için kullandığı ve non-lineer sistemleri simüle etmeye yarayan denklemleri kullanılmıştır. Bu denklemler sözde rasgele sayılar üretmek için kullanılmıştır. Modern şifreleme standartlarına uygun hale gelmesi , offline ve online durumlarda da kullanılabilmesi için Kaotik sistemler le RSA şifreleme algoritması beraber kullanılarak yeni bir şifreleme tekniği geliştirilmiştir. Güvenilirliği yüksek olan kaotik sistemlerle şifreleme algoritması RSA şifreleme algoritması ile birleştirilerek daha da güvenlikli hale getirilmiştir.

Çalışmanın uygulama kısmında kullanıcıların ihtiyaçları göz önünde bulundurularak hem kendi bilgisayarlarında güvenle kullanmaları,hem de internet üzerinde güvenle dosya gönderebilmeleri sağlanmıştır. Uygulama Borland Delphi 6.0 kullanarak programlanmıştır.

Anahtar Kelimeler:

Şifreleme, şifre çözme, açık anahtar, gizli anahtar, Kaos,Non-Lineer Sistemler,RSA, kriptografi

ABSTRACT

Master of Science Thesis

USE OF CHAOTIC SYSTEM
IN CRYPTOGRAPHY

OĞUZ FINDIK

Selcuk University
Graduate School of Natural and Applied Sciences
Department of Computer Engineering

Supervisor: ***Assoc.Prof.Dr.Şirzat KAHRAMANLI***

2004, *86 Pages*

Jury:

Prof. Dr. Ferruh YILDIZ
***Assoc.Prof.Dr.*Şirzat KAHRAMANLI**
***Assoc.Prof.Dr.*Ahmet ARSLAN**

Recently Internet has grown drastically and been used widely. Internet is naturally is open public network so it makes people all around globe get connected each other easily but also puts a lot of danger security of data which circulating in one computer to another .

In order to make internet a place where business can easily blossom, and people send emails each other securily and buy online without any doubt about what is going to happen my credit informaton, the data protection which flowing every day through one line to another must be secured.

At this point encryption comes to play very importan role. The encryption is a tool that is used to provide secure network data traffic and file storage. The more people use computer and computers get connected to each other the more bad-intended (hackers) users want to access private data. Because this reason the encryption is gaining importance every day.

In this work chaotic systems and RSA encryption algorithm are combined in order to develop an encryption algorithm which accomplishes the modern standards. In order to create chaotic map E.Lorenz's weather forecast' equations which are used to simulate non-linear systems. This equation can be used to generate random numbers. In order to achieve up-to-date standarts and use online and offline status a new encryption tecnique that combines chaotic sytems and RSA enryption algrorihtm has been developed. The combination of RSA algrorithm and chaotic systems makes very strong encryption system.

In the application part of the work, the encryption implementation was developed to satisfy the user needs. Public or private key can be given the program as a parameter. The application was developed by using Delphi 6.0 enterprise edition.

Key Words:

Encryption, decryption, public key, private key, Chaos,non-lineer systems,RSA, cryptography

ÖNSÖZ

İnternet bir çok bilgisayar sisteminin birbirine bağlı olduğu dünya çapında yaygın olan ve sürekli büyüyen bir iletişim ağıdır. İnternet'in ortaya çıkmasının önemli sebeplerinden birisi bilgiyi saklamak, paylaşmak ve ona kolayca ulaşma isteğidir. İnternet sayesinde insanlar birbirleriyle iletişim kurabilme imkanı bulabilmektedirler. Bu iletişim kurulurken güvenlik önlemleri almadan önemli bilgi sistemlerinin İnternet'e açılması büyük bir risk teşkil etmektedir. Ancak gerekli önlemler alındığında bu riskleri azaltmak mümkündür. Bu önlemlerden birisi de bilginin şifrelenerek (encryption) iletilmesidir. Şifreleme/deşifreleme (encryption-decryption) bir bilgisayar ağında veya kişisel bilgisayarlarda haberleşme ya da dosya güvenliğini sağlamak için kullanılır. Bu nedenle günümüzde bilgisayarlarda ya da bilgisayar ağlarında şifrelemenin önemi gün geçtikçe artmaktadır.

Tez çalışmamda beni teşvik eden ve çalışmam boyunca değerli öneri ve yardımlarıyla bana destek olan sayın hocam **Doç.Dr.Şirzat KAHRAMANLI'ya** teşekkürlerimi sunarım. Ayrıca tez konusu seçmemde ve sonraki aşamalarda yardımlarından dolayı sayın hocam **Doç.Dr. Ahmet ARSLAN**'a, şifreleme konusunda yardımlarını esirgemeyen **Arş.Gör. Halife KODAZ'a , Ali ÇETİNKAYA'ya ve** manevi desteklerinden dolayı **aileme** ve **Selçuk Üniversitesi Bilgisayar Mühendisliği Bölümü Personeline** teşekkürü bir borç bilirim.

İÇİNDEKİLER **Sayfa No**

ÖZET I
ABSTRACT III
ÖNSÖZ V
İÇİNDEKİLER VI
ŞEKİLLER VIII
TABLOLAR VIII
1. GİRİŞ 1
2. KAYNAK ARAŞTIRMASI 3
3. MATERYAL VE METOD 7
4. KRİPTOGRAFİ 8
4.1. Şifrelemenin Temel Elemanları 8
4.2. Gizli Anahtar (Simetrik) Yöntemleri 13
4.2.1. Geleneksel şifreleme teknikleri 14
4.2.2. Klasik şifreleme teknikleri 16
4.2.2.1. Yerine koyma teknikleri 16
4.2.2.1.1. Sezar şifreleme algoritması 16
4.2.3. Modern Şifreleme Teknikleri 18
4.2.3.1. Bit-bit/byte-byte şifreleme ve blok-blok şifreleme algoritmaları 18
4.2.3.2. Veri şifreleme standardı (DES) 19
4.2.3.3. Uluslar arası veri şifreleme algoritması (IDEA) 20
4.2.3.4. Blowfish 21
4.3. Açık Anahtar (Asimetrik) Şifreleme Yöntemleri 22
4.3.1. Asimetrik şifrelemede bilgi güvenliği 22
4.3.2. Asimetrik şifreleme yönteminin genel prensipleri 27
4.3.3. Asimetrik şifreleme yönteminin uygulamaları 32
4.3.3.1. RSA algoritması 33
4.3.3.1.1. RSA algoritmasının önemli özellikleri 40
4.3.4. Geleneksel şifreleme ile genel anahtarlı şifreleme farkları..... 41
4.4. Şifrelemede Kullanılan Anahtar Boyutları 41
4.4.1. Kriptografik sistemlerin güvenilirliği 44
4.4.2. Şifre çözme algoritmalarının analizi..... 46
4.4.2.1. Sadece şifreli metin saldırısı 47
4.4.2.2. Bilinen düz metin saldırısı 47
4.4.2.3. Seçilmiş düz metin saldırısı 48
4.4.2.4. Ortadaki adam saldırısı 48
5. KAOS 49
5.1. Lorenz Çekicisi..... 53
5.2.Nonlineerlik ve Türbülans..... 59
5.2.1.Kesir boyut ve Fraktal Geometri..... 61
5.2.2.Dalgalanmalı Düzen..... 63
6. UYGULAMA 65
6.1.Kaotik sistemler de veri şifreleme ve şifre çözme..... 65
6.2. Şifreleme 69
6.3.Şifre Çözme 70
6.4.Grafik Karşılaştırma 72
7. SONUÇLAR ve ÖNERİLER..... 74
8. KAYNAKLAR 76

EK-A ÇOK-BOYUTLU KAOTİK SİSTEMLER İLE RASGELE SAYILAR ÜRETEN ALGORİTMALAR 79
A.1 Chua Sistemini Kullanarak Rasgele Sayılar Üreten Algoritma……. 79
A.2 Lorenz Sistemini Kullanarak Rasgele Sayılar Üreten Algoritma…….. 83
EK B RSA Genel Anahtar Sistemi…… 86

ŞEKİLLER **Sayfa No**

Şekil 4.1.	Açık ağdan ileti gönderimi ……	8
Şekil 4.2.	Şifreleme ve şifreyi çözme işlemleri …….	8
Şekil 4.3.	Bir mesajın dinlenmesini önlemek için şifreleme kullanılması….	10
Şekil 4.4-a.	Tek anahtar ile şifreleme ve şifre çözme ……...	11
Şekil 4.4-b.	İki farklı anahtar ile şifreleme ve şifre çözme ……...	11
Şekil 4.5.	Geleneksel Şifreleme Sistem Modeli …….....……	15
Şekil 4.6.	Örnek Sezar şifresi uygulaması …….	17
Şekil 4.7.	Kullanıcı anahtar çiftlerinin bulunabileceği ortamlar..……..	24
Şekil 4.8.	Şifreli mesaj gönderilmesi ve alınması….. ……...	25
Şekil 4.9.	Sayısal imzalı mesaj gönderilmesi ve alınması…….	26
Şekil 4.10.a	Genel-anahtarlı Kriptosistem:Gizlilik……..…….	29
Şekil 4.10.b	Genel-anahtarlı Kriptosistem:Onaylama…….	29
Şekil 4.10.c	Genel-anahtarlı Kriptosistem:Gizlilik ve Onaylama……	29
Şekil 5.1.	Lorenz'in hava tahminlerini yapmak için kullanığı formullerin sonucunda elde edilen grafikler ……..…….....	52
Şekil 5.2.	Artan zamana göre X ve Y, Y=-7.11, Z=18.99 noktasında sarmal bir şekil……...	55
Şekil 5.3	R sabit değerinin değiştirilmesiyle elde edilen şekil……..	55
Şekil 5.4.	X'in zamana göre değişim grafiği……..	56
Şekil 5.5	X'in farklı değerler için elde edilen grafiği……	57
Şekil 5.6	Başlangıç değerlerinin değiştirilmesiyle elde edilen grafik……...	58
Şekil 5.7.	Başlangıç değerlerinin değiştirilmesiyle elde edilen grafik……...	58
Şekil 5.8.	Kelebek Etkisi……	60
Şekil 5.9	Koch'un kar taneleri……...	61
Şekil 5.10	Mandelbrot Kümesi……	63
Şekil 6.1.	Şifreleme…….....	69
Şekil 6.2	Şifre Çözme……...	70
Şekil 6.3.	Grafik Oluşturma……..	71
Şekil 6.4	Grafik Karşılaştırma…….	72
Şekil 6.5	Anahtar Oluşturma……...	73
Şekil 6.6.	Mod Bulma……	73

TABLOLAR **Sayfa No**

Tablo 4.1.	Geleneksel şifreleme ve genel anahtarlı şifreleme karşılaştırması …..	41
Tablo 4.2.	Farklı anahtar boylarında şifre kırma süreleri …….	42

1. GİRİŞ

Son yıllarda internet kullanımının yaygınlaşması bir takım güvenlik sorunlarını da beraberinde getirmiştir. Bunun başlıca sebebi, İnternet'in açık bir sistem olması ve üzerinde dolaşan verinin istenmeyen kişiler tarafından elde edilmesine uygun olmasıdır.

İnternet'te yollanan veri paketleri birçok halka açık ağdan geçer, bu da bu paketlere ulaşmayı mümkün kılar. Bu durum son derece gizli bilgiler internet 'te iletilirken, önemli bir kaygı halini alır. Bu tür bilgileri korumak mümkün olmadıkça, internet, iş yapmak veya kişisel yazışmalarda bulunmak için asla güvenli bir yer olmayacaktır.

Bilgi güvenliği; başkası tarafından dinlenme, bilginin değiştirilmesi, kimlik taklidi gibi tehditlerin ortadan kaldırılması ile sağlanır ve bu amaçla kullanılan temel araç şifrelemedir.

Ağ ve haberleşme güvenliği için en önemli araç şifrelemedir. Kriptografi bilgi güvenliğini inceleyen ve anlaşılabileni anlaşılamaz yapan bir bilim dalıdır. Güvenilirlik, veri bütünlüğü, kimlik doğrulama gibi bilgi güvenliği konularıyla ilgilenen matematiksel yöntemler üzerine yapılan çalışmalar şifrelemenin önemli konularıdır.

Kriptografinin tam olarak Türkçe karşılığı "şifre yazım"dır. Kriptografi yunanca gizli anlamına gelem "crypt" ve yazı anlamına gelen "graph"'dan kelimelerinden türetilmiştir. Kriptoloji ise şifre bilimidir. Şifre kelimesinin kökü ise Fransızcıdaki "chiffre" yani sayı kelimesinden gelmektedir.

Veri iletişimin gizliliğini ve güvenliğini sağlamak için işlerini ciddiye alan bütün kurum ve kuruluşların şifreleme(kriptografi) kullanması gereklidir.

Bilgi toplumlarına bakıldığında, teknolojinin de yardımıyla, bilgisayar ortamındaki bilgiler başkalarının eline geçebilmektedir. Kriptografi dijital dünyadaki

kişileri bu durumdan koruyan bir araçtır. Kriptografiyi öğrenmek ve onun modern toplumlara sağladığı avantajlardan yararlanmak hem kişisel gizlilik hem de elektronik dünyada güvenlik için kaçınılmazdır.

Kriptografi, yetkili olmayan kişilerin görmesini istemediğiniz e-maili, mesajı veya diskteki herhangi bir dosyayı şifrelemeyi sağlayarak gizliliği korur.

Kriptografi genel olarak şu ana konularla ilgilenir:

- **Gizlilik:** Bilgi istenmeyen kişiler tarafından anlaşılamamalı.

- **Bütünlük:** Bir iletinin alıcısı bu iletinin iletim sırasında değişikliğe uğrayıp uğramadığını öğrenmek isteyebilir; davetsiz bir misafir doğru iletinin yerine yanlış bir ileti koyma şansına erişmemelidir. Saklanan veya iletilmek istenen bilgi farkına varılmadan değiştirilememeli.

- **Reddedilemezlik:** Bilgiyi gönderen kişi, daha sonra bilgiyi kendisinin gönderdiğini inkar edememeli.

- **Kimlik belirleme:** Gönderen ve alıcı, birbirlerinin kimliklerini doğrulayabilirler. Davetsiz bir misafir başkasının kimliğine bürünme şansına erişmemelidir.

2. KAYNAK ARAŞTIRMASI

Şifreleme çok eski tarihlerden beri insanoğlunun ilgisini çekmiş bir konudur. Öncellikle askeri alanda sık kullanılmaya başlanmış, daha sonra bilgisayar teknolojisinin gelişmesi ile birlikte geniş bir uygulama alanı bulmuştur. Güvenli veri iletişimi yapabilmek için çeşitli standartlar ve algoritmalar geliştirilmiştir. Günümüzde de bu çalışmalar devam etmektedir.

Bu tez çalışması sırasında şifreleme ilgili sahalarda çalışan kişi, kurum ve organizasyonların yaptıkları çalışmalar ve bu çalışmaların sonuçları incelenmiştir.

Aşağıda bu çalışmalarla ilgili bilgiler verilmiştir:

(1960) Basit matematik denklemleri kullanılarak şiddetli sistemlerin(şelale gibi) simüle edilmesiyle kaos tanımlanmaya başlanmıştır.

E.Lorenz(1963) Hava değişimlerinin başlangıç değerlerine hassasiyetle bağlı olduğunu keşfetmiştir.

J.Ford ve G.Casati(1977)Kaos adı verilen bu bilim dalında ilk kez bir konferans düzenlediler.

Apohan (1993), tez çalışmasında şifreleme konusunu genel hatlarıyla incelemiştir.

Erhan (1993), tez çalışmasında açık anahtarlı şifreleme yöntemi olan RSA algoritmasını kullanan şifreleme/ deşifreleme yazılımı geliştirmiştir.

Stinson (1995), şifreleme kavramını teorik ve pratik olarak anlatan bir kitap yazmıştır. http://www.cacr.math.uwaterloo.ca/~dstinson/CTAP.html sayfasında kitapla ilgili bilgiler bulunmaktadır.

Schneier (1996), kriptografik algoritmaların, protokollerin ve yöntemlerin uygulamasına ilişkin ayrıntılı bilgi veren bir kitap yazmıştır. Konuyla ilgilenenlerin okumasında büyük fayda vardır. http://www.counterpane.com/applied.html sayfasında kitapla ilgili bilgiler bulunmaktadır

Yiğit(1997) tez çalışmasında çok boyutlu kaotik sistemler kullanarak verilerin şifrelenmesi ve bu sistemlerin rasgeleliğini incelemiştir.

Develi(1997), tez çalışmasında güvenli haberleşme için kaotik işaretleri ve sistemleri incelemiştir.

Gül (1997), tez çalışmasında açık anahtarlı şifreleme yöntemi olan RSA algoritması ile C programlama dilinde uygulama geliştirmiştir.

Naccache (1999), çalışmasında RSA asimetrik şifreleme sistemine karşı yapılan saldırıları incelemiştir.

Golic (1999), çalışmasında rastgele erişimli dosyalarda güvenli ve verimli şifreleme/deşifreleme işlemlerini incelemiştir.

Wright (1999), çalışmasında DES algoritmasından AES algoritmasına kadar şifreleme konusunda olan gelişmeler kronolojik sırada incelemiştir.

Aslan (1999), tez çalışmasında sayısal imzalama sistemlerini incelemiştir.

Saka (2000), tez çalışmasında bilgisayar ağ güvenliği ve şifreleme üzerine çalışma yapmıştır.

Nan , Wong , Tsang ,Shi(2000) Lineer senkronize edilmiş kaotik haritalara dayalı güvenli sayısal haberleşmeyi incelemiştir.

Jakimoski, Kocarev(2001) Kaos tabanlı şifreleme algoritmaları üzerine bir çalışma yapmıştır.

Chang, Hwang ve Chen (2001), makalesinde görüntü dosyaları üzerinde şifreleme işlemi yapan bir algoritma incelemiştir.

Peyravian, Matyas, Roginsky ve Zunic (2001), çalışmasında RSA algoritmasında kullanmak için gerekli olan asal sayıların üretilmesi için alternatif bir metot geliştirmiştir.

Torrubia, Mora ve Martı (2001), çalışmasında elektronik ticarette şifreleme kavramı ve sayısal ortamda kişilerin haklarına ülkelerin bakış açılarını incelemiştir

Schmitz(2001), Şifrelemede kaotik dinamik sistemlerin kullanımını incelemiştir.

Qurban A. Memon(2002), Seknronize kaos'u ağ güvenliğinde kullanmak için bir çalışma yapmıştır.

Palacios ,Juarez(2002),Kaos sinyalleri ile şifrelemenin gerçekleştirilmesi üzerine bir çalışma yapmıştır.

Garcia, Jimenez (2002) Kaotik haritalar yönetimiyle haberleşmeyi incelemiştir.

Kodaz(2002); Tez çalışmasında şifreleme algoritmalarını araştırmış ve RSA şifreleme algoritması kullanarak verilerin şifrelenmesini incelemiştir.

Wu, Hu, Zhang(2004); yaptıkları yayında chaos temelli haberleşmenin nasıl daha iyi hale getirileceğini araştırmışlardır

Kaynak araştırması sonucunda güvenli bir veri iletimini gerçekleştirebilmek amacıyla kaotik modelle şifreleme metodu ile RSA şifreleme metodu bir arada kullanılmıştır. Uygulamalar Delphi 6.0 programlama dilinde hazırlanmıştır. Uygulama yazılımını ekran görüntüleri tez raporu içerisinde verilmiştir.

Uygulama yazılımlarının kaynak kodları ve kurulum dosyaları CD içerisinde ek olarak verilmiştir.

3. MATERYAL VE METOD

Bu çalışmada şifrelemenin genel mantığı, şifrelemeye neden gerek duyulduğu, kaotik modelle şifreleme mantığı ve bu şifrelemenin nasıl daha iyi duruma getirilebileceği incelenmiştir.

Çığ gibi büyüyen bilgi teknolojisinde bilginin güvenli olarak saklanması ya da iletilmesi giderek önem kazanmıştır. Bu sebepten dolayı şifreleme geniş bir uygulama alanı bulmuştur. Uluslar arası alanda şifreleme tekniklerini tespit edebilmek ve standartlar oluşturmak için çeşitli kuruluş ve organizasyonlar ortaya çıkmıştır (NSA, ANSI, NIST vb).

Bu çalışmada güvenli veri iletimini gerçekleştirebilmek için E.Lorenz'in lineer olmayan denklemleri kullanarak rasgele sayı dizileri elde edilmiş ve bu sayı dizileriyle veriler şifrelenmiş ve RSA algoritması ile birleştirilerek online ve offline ortamlarda kullanılabilecek hale getirilmiştir. Bu algoritma Delphi 6.0 kullanılarak programlanmıştır.

4. KRİPTOGRAFİ

4.1. Şifrelemenin Temel Elemanları

Elektronik iletişim, günümüzde kağıt üzerinde yazı yazarak yapılan her türlü iletişimin yerine geçmeye başlamıştır. Kişi/kuruluş/toplumların, özel/kamusal/resmi haberleşmelerini elektronik iletişim ağları üzerinden yapabilmeleri, açık ağlar üzerinden iletilen bilginin güvenliği ve güvenilirliğiyle yakından ilgilidir.

Bir göndericinin, bir alıcıya açık ağlar üzerinden bir ileti gönderdiği zaman, açık ağlardan gönderilen iletiler üçüncü şahıslar tarafından dinlenme ve değiştirilme tehdidi altındadırlar (Şekil 4.1).



Şekil 4.1 Açık ağdan ileti gönderimi

Burada söz konusu ileti düz metindir. Bazı kullanımlarda *plaintext* adı da verilir. Bir iletinin içeriğini saklamak üzere yapılan gizleme işlemi de şifrelemedir (*encryption*). Bu işlem düz metini şifreli metine dönüştürür. Bilginin içeriği başkalarının anlamayacağı hale gelir. Bu bilgi, bir yere iletilmek amacıyla şifrelenen bir mesaj veya saklanmak amacıyla şifrelenen bir bilgi olabilir. Şifrelenmiş bir ileti şifreli metindir (*ciphertext*). Şifreli metini düz metine geri çevirme işlemi şifre çözümüdür (*decrypt*). Bu işlemler Şekil 4.2'de gösterilmektedir.



Şekil 4.2. Şifreleme ve şifreyi çözme işlemleri

İleti güvenliğini sağlama bilimi *kriptografidir*. Matematiğin hem şifre bilimi hem de şifre analizini kapsayan dalı *kriptolojidir* ve şifre bilimciler tarafından icra edilir.

Düz metin bir metin dosyası, resim, ses, görüntü dosyası yani sayısal ortamındaki her şey olabilir. Bilgisayar için düz metin sadece ikili (binary) bir veridir. Düz metin iletim veya depolama amacıyla tasarlanabilir. Her iki durumda da, düz metin şifrelenecek iletidir.

Şifreli metin de ikili veridir, bazen düz metin ile aynı boyutta, bazen daha büyüktür. Şifreleme işlemi şifreli metini üretmek üzere düz metin üzerinde gerçekleştirilir.

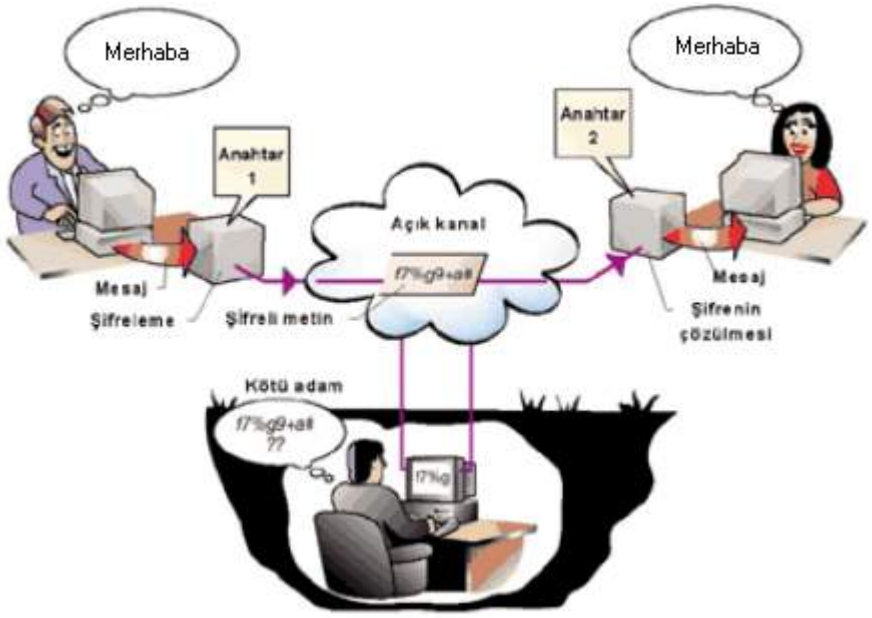
Kriptografik algoritmalar adı da verilen şifreleme algoritmaları, şifreleme ve şifre çözümü için kullanılan matematiksel işlemlerdir. Eğer bir algoritmanın güvenliği bu algoritmanın çalışma biçimini gizlemeye dayalıysa, bu bir sınırlandırılmış algoritmadır. Geçmişte ilgilenilen kriptografi algoritmaları algoritmanın gizliliğine dayanmaktaydı. Sınırlandırılmış algoritmalar günümüzün şartlarına pek uymamaktadır; bir gruba ait kullanıcılar bunları kullanamamaktadır, çünkü gruptan bir kullanıcının her çıkışında geri kalan herkesin başka bir algoritmaya geçmesi gerekmektedir. İçlerinden birisi yanlışlıkla ya da bilinçli olarak kullanılan algoritmayı açığa vurduğunda, diğer herkesin algoritmalarını değiştirmeleri gerekmektedir. Daha da kötüsü, sınırlandırılmış algoritmalar kalite kontrolüne ve standardizasyona olanak tanımamaktadır. Her bir grup kullanıcının kendisine ait bir algoritması olmalıdır. Bu tür bir grup hazır şifre çözüm anahtarının yazılım veya donanım ürünlerini kullanamaz; davetsiz bir misafir aynı ürünü alıp algoritmayı öğrenebilir. Kendi algoritmalarını ve gerçekleştirimlerini kendileri yazmaları gerekir.

Bu tip ciddi eksiklikleri olmasına rağmen, sınırlandırılmış algoritmalar düşük güvenlik gerektiren uygulamalarda sık sık kullanılmaktadır. Kullanıcılar

sistemlerinde bulunan güvenlik sorunlarının ya farkında değildir ya da bunları önemsememektedir.

Günümüz kriptografisi bu sorunu bir anahtar ile çözmektedir. Bu anahtar çok çeşitli değerler alabilen herhangi bir anahtar olabilir. Anahtarın alabileceği olası değerler genişliğine *anahtar uzayı* denir.

Günümüzde kullanılmakta olan modern ve güçlü şifreleme algoritmaları artık gizli değildir. Bu algoritmalar güvenliklerini kullandıkları farklı uzunluk ve yapılardaki anahtarlarla sağlarlar. Bütün modern algoritmalar şifrelemeyi ve şifre çözmeyi kontrol için anahtarları kullanır. Bir anahtar ile şifrelenmiş bilgi, kullanılan algoritmaya bağlı olarak, ilgili anahtar ile çözülebilir. Genel olarak anahtarın kullanımı şu şekildedir (şekil 4.3):



Şekil 4.3. Bir mesajın dinlenmesini önlemek için şifreleme kullanılması.

Kullanıcı bir mesajı (m) göndermeden önce bir anahtar (k1) kullanarak şifreler. Şifreli metin (c) yasadışı dinleyicilere açık olan bir kanaldan gönderilir. Mesajı okumak için alıcı bir anahtar (k2) kullanarak şifreyi çözer ve m mesajını elde eder. Aktif düşmanlar araya girip iletişimi dinleyebilir.

Eğer k1 ve k2 eşitse, sistem *simetriktir*. Aksi takdirde bu sistem *asimetrik* olarak nitelenir. Güvenliğin garantilenmesi için k2 her zaman gizli olmalıdır, ancak k1'i kullanarak k2'yi elde etmek mümkün olmadığı sürece k1 açıklanabilir. Bu durumda sisteme açık anahtarlı sistem (public key system) adı verilir.

Açık anahtarlı sistemler pek çok ilginç olanaklar sunar; örneğin herkes online bir mağazaya mağazanın açık k1 anahtarını kullanarak şifrelenmiş bir kredi kartı numarası gönderebilir. k2 anahtarını sadece mağaza bildiği için, kartın numarasını sadece mağaza öğrenebilir. Eğer simetrik sistem kullanılsaydı, mağaza potansiyel müşterilerinin her biriyle önceden ve gizlice ayrı ayrı anahtarlar belirlemek zorunda kalırdı. Açık anahtarlı sistemlerin güvenliği her zaman belirli matematiksel problemleri çözmenin zorluğuna dayanır, simetrik sistemler daha çok tek kullanımlık, geçici yapıdadırlar. Açık anahtarlı sistemlerin en büyük dezavantajı matematiksel yapıları nedeniyle simetrik sistemlerden daha yavaş olmalarıdır; özellikle açık anahtarlı sistemlerdeki anahtarların boyutları simetrik sistemlerin anahtarlarının boyutlarından çok daha büyüktür. Kısaca, kullanılacak şifreleme yöntemi gerçekleştirilecek uygulamaya bağlı olarak seçilir. (Şekil 4.4-a, şekil 4.4-b)



Şekil 4.4-a Tek anahtar ile şifreleme ve şifre çözme



Şekil 4.4-b İki farklı anahtar ile şifreleme ve şifre çözme

Algoritmalardaki bütün güvenlik anahtara (veya anahtarlara) dayalıdır, hiçbiri algoritmanın ayrıntılarında yer almaz. Bu, algoritmanın yayınlanabildiği ve incelenebildiği anlamına gelir. Bu algoritmayı kullanan ürünler seri üretilebilir. Bir

davetsiz misafirin sizin algoritmanızı bilmesi önemli değildir; sizin özel anahtarınızı bilmedikçe, o şahıs iletilerinizi okuyamaz.

Bir şifre sistemi algoritmalardan ve olası bütün düz metinlerden, şifreli metinlerden ve anahtarlardan oluşur.

Şifreleme sistemleri genel olarak birbirinden bağımsız üç boyutta incelenir:

- Orijinal metini şifreli metine dönüştürme operasyonları: Bütün şifreleme algoritmaları temelde iki genel prensibe dayanır. Yerine koyma, orijinal metindeki her eleman (bit,harf,bit veya harf grupları) yerine başka bir eleman konur. Yer değiştirme, orijinal metindeki her bir elemanın yeri değiştirilir. Burada temel gereklilik hiç bir elemanın kaybedilmemesidir.

- Kullanılan anahtarların sayısı: Bütün gönderici ve alıcılar aynı anahtarı kullanırsa sistem; simetrik, tek anahtarlı, gizli anahtarlı veya geleneksel şifreleme olarak bilinir. Eğer gönderici ve alıcı farklı anahtarlar kullanırsa bu asimetrik; iki anahtarlı yada kişisel anahtarlamalı şifreleme olarak bilinir.

- Orijinal metine uygulanan süreçler: Bir blok şifreleme(Block cipher) işlemi, bir periyotta bir girdi bloğu için bir çıktı bloğu üreten işlemdir. Arka arkaya gelen bir veriyi şifreleme (Stream cipher) işlemi bir periyotta akan arka arkaya gelen girdi veriler için bir eleman üreten işlemdir.

Şifreleme algoritmaları anahtar kullanma yöntemlerine göre genel olarak iki kategoriye ayrılmaktadır. Bu yöntemler:

- Gizli-Anahtar (Simetrik) yöntemleri (Geleneksel şifreleme sistemleri)

- Açık-Anahtar (Asimetrik) yöntemleri (Açık anahtar şifreleme sistemleri)

Geleneksel şifreleme sistemleri şifreleme yaparken gizli bir anahtar kullanırlar. Bu anahtarın dağıtımı önemli bir sorundur. Bu sorun açık anahtar şifreleme sistemleri ile çözülebilir.

4.2. Gizli Anahtar (Simetrik) Yöntemleri

A ve B mesajlarını diğerleri için anlaşılmaz kılacak bir yöntem üzerine gizlice anlaşırlar. Örneğin, bütün kelimelerin tersten yazılacağı (MERVE yerine EVREM), veya her harften sonra rasgele bir harf konulacağı kararlaştırılabilir (MERVE yerine MŞEARKVİER). Yöntemin bulunması durumunda bütün iletişimler çözülecektir. Farklı kişilerle iletişim kurmak istenirse, her kişi ile farklı yöntemler üzerinde anlaşılması gerekir. Bir çok kişinin şifreyi bilmesi gerekiyorsa, güvenlik için sık sık yöntemin değiştirilmesi gerekebilir. Yeni yöntemler bulmak ise sanıldığından daha zordur.

Bu aşamada, temel işleyişi sabit olan ama parametreli bir yöntem gerekir. Artık birden fazla kişi ile temel mekanizma üzerinde, gizliliğe gerek duymadan anlaşabilirsiniz; gizli olarak anlaşmanız gereken tek kısım, şifreleme yönteminin değişken kısmıdır. Bu değişkene şifreleme anahtarı denir. Bu çeşit yöntemlere **Gizli Anahtar Anlaşmalı Açık Yöntemler** tanımı getirilebilir. Bu tür yöntemlere **simetrik şifreleme** adı verilir.

Gizli anahtar algoritmaları hem şifreleme hem de şifre çözmek için aynı anahtarı kullanırlar. Bu durum veri şifreleme için matematiksel açıdan daha az problem çıkaran bir yaklaşımdır ve çok kullanılan bir yöntemdir.

Tek anahtarlı, gizli anahtarlı veya özel anahtarlı algoritmalar da denilen bu algoritmalar, güvenli bir şekilde iletişim kurmalarından önce gönderici ile alıcının özel anahtar olarak adlandırılan bir anahtar üzerinde uzlaşmalarını gerektirir. Bir simetrik algoritmanın güvenliği anahtarda yatar; anahtarı herkesin bilmesi, isteyen herkesin iletileri şifreleyebileceği ve şifreleri çözebileceği anlamına gelir. İletişimin gizli kalması için anahtarın da gizli kalması gereklidir.

Simetrik algoritmalar iki kategoriye bölünebilir. Kimileri düz metin üzerinden zaman başına (belli bir anda) tek bit (veya bazen sekiz bit) olarak çalışır; bu tür algoritmalara "stream algoritmalar" denir. Diğerleri düz metin üzerinden bit grupları halinde çalışır. Bit gruplarına bloklar denir, algoritmalara da blok algoritmalar denir. Modern bilgisayar algoritmaları için, tipik bir blok büyüklüğü 64 bittir.

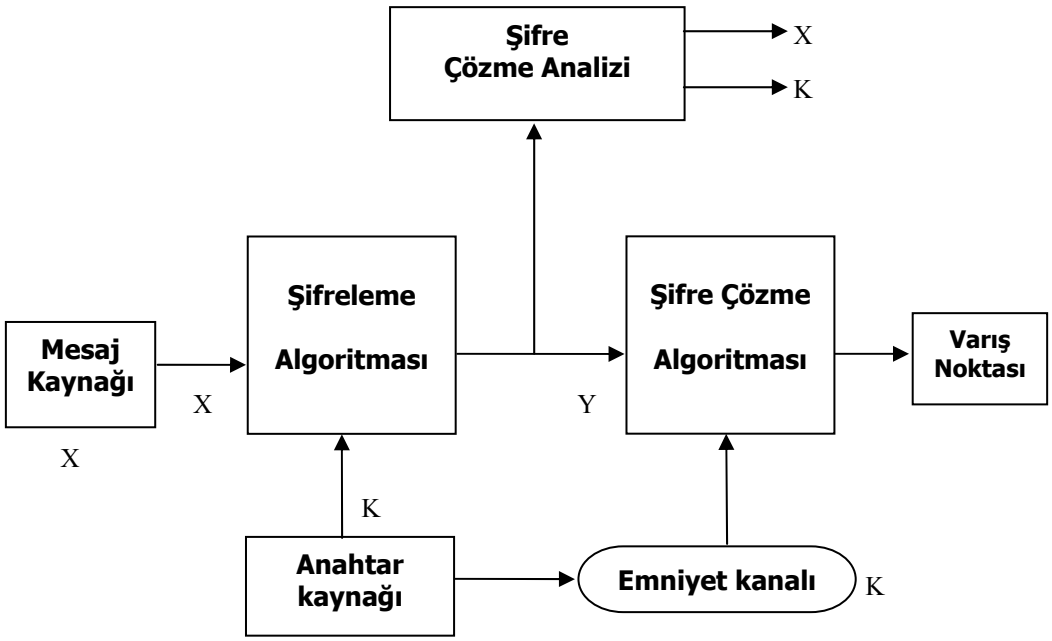
Simetrik şifreleme algoritmaları; geleneksel şifreleme teknikleri, klasik şifreleme teknikleri ve modern şifreleme teknikleri başlıkları altında incelenmiştir.

4.2.1. Geleneksel şifreleme teknikleri

Bu modelde mesaj şifrelenirken orijinal (anlaşılır) mesaj (*plaintext*) şifreli (anlaşılmaz) bir hale (*ciphertext*) dönüştürülür. Bu şifreleme süreci (*encryption*), bir algoritma ve bir anahtardan oluşur. Anahtar orijinal mesajdan bağımsız bir değerdir. Algoritma aynı oturumda kullanılan belirli anahtara bağlı olarak bir sonuç üretir. Anahtarı değiştirmek, algoritmanın çıktısını da değiştirir.

Şifreli mesaj orijinal (anlaşılır) mesaja dönüştürme(*decryption*) algoritmasına gönderilerek anlaşılır hale gelir. Şifreli metinin orijinal metine dönüştürme işlemi, şifreleme aşamasında kullanılan anahtar aracılığıyla gerçekleşir.

Geleneksel şifreleme güvenliği, çeşitli faktörlere bağlıdır. İlk olarak şifreleme algoritmasının çözülemeyecek kadar güçlü olması gerekir. Bunun ötesinde şifreleme güvenliği algoritmanın gizliliğine değil, anahtarın gizliliğine bağlıdır. Başka bir deyişle algoritma gizli tutmamalı sadece anahtar gizli tutulmalıdır. Geleneksel şifrelemenin çok yaygın kullanılmasını mümkün kılan bu özelliğidir. Algoritma gizliliğine gerek yoktur. Geleneksel şifrelemenin bu kullanımı güvenlik problemlerini anahtar gizliliği ile güvence altına alır.



Şekil 4.5 Geleneksel Şifreleme Sistem Modeli

Şekil 4.5'de geleneksel şifreleme sistem modeli gösterilmiştir. Bir mesaj kaynağı olan X, anlaşılır mesajı üretir. Şifreleme için bir anahtar belirlenir. Orijinal mesaj için bir anahtar üretilmişse bu anahtar mesaj kaynağından bir güvenlik kaynağıyla karşı tarafa ulaştırılmalıdır.

X mesajı ve K şifreleme anahtarı şifreleme algoritmasına girdi olarak alınır. Bunun sonucunda şifreleme algoritması (E) nin ürettiği şifreli metin ortaya çıkar (Y).

$$\mathbf{Y=E_K(X)}$$

Bu notasyon Y şifreli metini E şifreleme algoritmasının bir fonksiyonu olduğunu gösterir. E şifreleme algoritması da K anahtarını kullanarak çalışır.

Art niyeti olan bir kişi anahtara sahip olarak gerekli dönüşümü yapabilir.

$$X=D_K(Y)$$

Y şifreli metinini gözlemleyen ancak K şifreleme anahtarı veya X mesajına ulaşamayan bir kişi X veya K' yı yada her ikisini de ele geçirmeye çalışır.

Farz edelim ki bu art niyetli kişi şifreleme algoritmasını (Encryption-E) ve deşifre algoritmasını (Decryption -D) bilsin. Bu kişinin ilgisi sadece orijinal mesaj

ise tahmini bir X mesajı oluşturarak X i elde etmeye çalışacak ve tahmini bir K şifreleme anahtarı oluşturup K anahtarını da elde etmeye teşebbüs edecektir.

4.2.2. Klasik şifreleme teknikleri

4.2.2.1 Yerine koyma teknikleri:

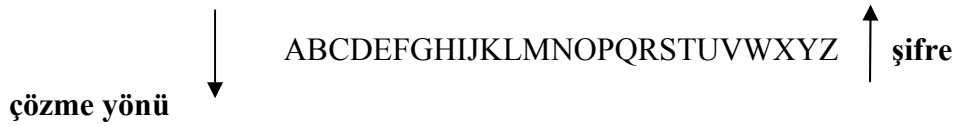
Yerine koyma tekniği; orijinal metindeki bir karakter yerine başka karakter yada karakterler koyma işlemidir. Orijinal metin nasıl karakterlerden oluşan bir yapıya sahipse yerine koyma tekniğinden sonra oluşacak şifreli metin de değişik karakterlerden oluşan bir yapıya sahip olacaktır.

4.2.2.1.1. Sezar şifreleme algoritması

Yerine Koyma algoritması ilk ve basit şekli ile eski Roma imparatoru *Julius Caesar* tarafından kullanılmıştır. Sezar algoritması orijinal metindeki her harf yerine o harften sonra gelen n.harfi yerine koymayı öngörür. n değeri anahtar görevi görmektedir. Alfabenin harfleri bir döngü oluşturacak şekilde tasarlanmıştır. Örneğin n=3 için Z nin yerine C gelir.

Sezar Şifresini kullanmanın basit bir yolu, alfabenin bütün harflerini bir kağıt şeridi üzerine yazıp, şeridin başını sonuna yapıştırmaktır. Gizli anahtar olarak (veya şifrenin değişkeni), 1 ile kullanılan alfabedeki harf sayısının 1 eksiği arasında bir n sayısı seçilir. Metinimizi şifrelemek için, harfleri teker teker alırız; şeridin üzerinde o harfin yerini bulup, sağa doğru n tane harf atlarız; şifreli metinimizde ise o harfin yerine bulduğumuz yeni harfi buluruz. Şifreyi çözmek için, aynı işlemi, şeridi aksi yönde çevirerek gerçekleştiririz. Şifreyi çözebilmek için, gönderici ile alıcı ortak bir n değerinde anlaşması gerekir.

Örnek olarak Latin alfabesiyle, **n=13** parametresiyle şifrelenmiş bir Sezar Şifresini şekil 4.6 'de gösterilmiştir.



Açık Metin :"BU YAZDIKLARIM ARAMIZDA KALSIN"

Şifrelenmiş Metin :"OH LNMQVXYNEVZ NENZVMQN XNYFVA"

Şekil 4.6 Örnek Sezar şifresi uygulaması

Sezar şifresini çözmek için n sayısını bulmak yeterlidir. n sayısının alabileceği farklı değerler sınırlıdır. n sayısı en fazla kullanılan alfabedeki harf sayısından bir eksik olacaktır yani 28 (Türkçe için). Bütün 28 ihtimal teker teker denenebilir. Bu işlem bilgisayarda sadece bir kaç mikro saniye sürer. Simetrik bir şifreyi çözmek için anahtar değerlerini teker teker deneme yöntemine, İngilizce *"Brute Force Search"* adı verilmektedir. Türkçe deneme-yanılma yöntemi denilebilir.

Kullanılan şifre çözme yöntemi bu problemin 3 önemli özelliğini kullanır:

- Şifreleme deşifreleme algoritmaları biliniyor.
- Denenebilecek sadece 28 anahtar vardır.
- Bu orijinal metinin dili biliniyor ve kolayca yorumlanabiliyor.

C = (P+K) mod 29

C : şifrelenmiş belge(ciphertext)

P : şifrelenmemiş belge(plaintext)

K : şifreleme anahtarı(simetrik)

Bu duruma göre:

P =ŞAKİR; K=6 için C=?

Ş-A-K-İ-R = 22-0-13-11-20

Harflerin alfabedeki yerlerine göre her harf için şifreleme algoritması K=6 için uygulanırsa:

P1= “Ş”;

C1= (P1+K) mod 29= (22+6)mod 29 =28= “Z”

P2= ”A”;

C2= (P1+K) mod 29= (0+6)mod 29 =6= “F”

...

C = ZFPOV

Şifrelenmiş belgeyi deşifre etmek için :

C1= “Z”

P1= (C1-K) mod 29= (28-6)mod 29 = 22= “Ş”

C2= “F”

P2= (C2-K) mod 29= (6-6)mod 29 = 0 = “A”

4.2.3. Modern Şifreleme Teknikleri

Modern şifreleme teknikleri blok şeklinde şifreleme ve deşifreleme yapmaktadır. Hemen hemen bütün simetrik blok şifreleme algoritmaları “*Feistel blok şifresi*” olarak bilinen bir yapı üzerine kurulmuştur. Bu sebepten dolayı öncelikle bit-bit ve blok-blok şifreleme yapan algoritmalar hakkında bilgi vermek konunun anlaşılması için faydalı olacaktır.

4.2.3.1. Bit-bit/byte-byte şifreleme ve blok-blok şifreleme algoritmaları

Her bir periyotta dijital veri akışında bir bit veya bir byte veri şifreleyen yöntemler *stream şifreleme* olarak bilinir. Bir blok şifreleme ise orijinal metin bloğunu, aynı uzunlukta bir şifreli metin bloğuna çevirir. Esasen bir blok 64 bitlik veri kullanır.

Blok şifrelemenin analiz edilmesine daha çok çaba harcanmıştır. Genel olarak blok şifreleyiciler, uygulamaların genişliği açısından stream şifreleyicilerden daha uygulanabilir görünmektedir.

Yaygın olarak kullanılan şifreleyicilerin çoğu blok şifreleyicilerdir. Blok şifreleyiciler sabit genişlikte (genellikle 64 bit) veri bloklarını diğer bir sabit genişlikli (yine 64 bit genişliğindeki) bloğa, anahtar tarafından seçilen bir fonksiyon kullanarak dönüştürürler. Eğer anahtar, girdi bloğu ve çıktı bloğunun hepsi de *n* bit ise, bir blok şifreleyici temel olarak *n*-bit tamsayılardan *n*-bit tamsayıların permutasyonlarına bir bire-bir eşleme (mapping) tanımlanır.

4.2.3.2. Veri şifreleme standardı (DES)

Çok yaygın kullanılan şifreleme tasarımları, 1977 yılında NBS firması tarafından oluşturulmuştur. Kurumun bugünkü adı ABD Ulusal Teknoloji ve Standartlar Enstitüsüdür (NIST). Bu kurumun geliştirdiği DES (Data Encryption standart) algoritmasında veriler 64 bitlik bloklar halinde ve 56 bitlik anahtar kullanılarak şifrelenir. Algoritma, 64 bitlik giriş bilgisini bir dizi işlemlerle yine 64 bitlik bir diziye dönüştürür. Aynı anahtarla aynı işlemler deşifreleme için kullanılır.

56 bit anahtar uzunluğunda bir anahtar kullanımında yaklaşık $7{,}2 \times 10^{16}$ adet anahtara denk düşen muhtemel 2^{56} adet anahtar vardır. Böylece bir deneme yanılma (brute force) saldırısı çok zordur.

Her simetrik şifreleme algoritmasında olduğu gibi algoritmaya iki giriş vardır, şifrelenecek orijinal metin ve şifreleme anahtarı. Bu durumda orijinal metin 64 bit uzunlukta bloklar halinde ve anahtar 56 bit uzunlukta olmalıdır. Anahtar uzunluğunun da 64 bit olması beklenir fakat 56 bitlik anahtar kullanılır. Diğer 8 bit, denetim biti olarak kullanılır (parity bitleri).

Algoritmada orijinal metin 3 safhadan geçmektedir. Bunlar başlangıç permutasyonu, 16 adet raund ve ters permutasyon safhalarıdır. İlk safha başlangıç permutasyonudur (IP Initial Permutation). Bu safhada 64 bitlik orijinal metin permutasyon fonksiyonuna girer ve metnin yeni formu elde edilir. İkinci safha birbirinin aynısı olan permutasyon ve yerine koyma fonksiyonlarından ibaret olan 16 adet raunddan oluşur. 16. raundun sonunda algoritmaya giren orijinal metin ve anahtarın bir fonksiyonu olan 64 bitlik bir veri elde edilmiş olur. Son safha olarak 64

bitlik şifreli metini oluşturmak için veriye başlangıç permutasyonunun tersi olan permutasyon fonksiyonu uygulanır.

4.2.3.3. Uluslar arası veri şifreleme algoritması (IDEA)

IDEA (International Data Encryption Algorithm - Uluslararası Veri Şifreleme Algoritması) Xuejia Lai ve James Massey tarafından İsveç Federal Teknoloji Enstitüsünde geliştirilmiş simetrik bir blok şifreleme algoritmasıdır. Algoritma 128 bitlik bir anahtar kullanır.

IDEA, çoğunluk tarafından yararlı kullanım ömrünü doldurduğu yada doldurmak üzere olduğu düşünülen DES'in yerini almak üzeredir ve son yıllarda önerilen çok sayıdaki geleneksel şifreleme algoritmalarından biridir. Uyarlama yönünden, IDEA bu önerilerin en başarılarından biridir.

- Tasarım İlkeleri: IDEA bir blok şifreleme algoritmasıdır. Veriyi 64 bitlik bloklar halinde şifreleyen 128 bitlik anahtar kullanır. Bunun aksine DES, 64 bitlik bloklar fakat 56 bitlik anahtar kullanmaktadır. IDEA tasarımında hedef, kriptografik gücü arttırmak ve uygulama kolaylığı sağlamaktır.
- IDEA Şifrelemesi: Bir çok şifreleme tasarımında olduğu gibi, şifreleme fonksiyonunda iki girdi vardır; şifrelenecek orijinal metin ve şifreleme anahtarı. Bu durumda orijinal metin 64 bit, anahtar ise 128 bit uzunluğundadır.

IDEA şifreleme algoritması 8 raunddan oluşur ve aşağıdaki operasyonlar uygulanır;

- Algoritma, girdiyi 16 bitlik alt bloklara böler. Her bir raundda, bu 16 bitlik bloklar girdi olarak alınır ve 16 bitlik çıktılar elde üretilir.
- Son dönüşümde, daha sonra birleşerek 64 bitlik bir şifreli metni oluşturan 4 tane 16 bitlik bloklar üretir.
- Her raundda 6 adet ve son dönüşümde 4 adet olmak üzere toplam 52 adet 16 bitlik alt anahtar kullanılır.

- 52 adet alt anahtar, 128 bitlik anahtardan oluşturulur.

Şifreleme operasyonları aşağıdaki adımlardan oluşur;

- **Bitwise exclusive-OR** : Bit-bit özel-VEYA operasyonu, bu operasyon "⊕" gösterilir.
- **Addition** : Word'lerin toplama alınması,bu operasyon "□" ile gösterilir.
- **Multiplication** : Bu operasyon "O" ile gösterilir.

4.2.3.4. Blowfish

Blowfish, Bruce Schneier (1993-1994) tarafından geliştirilmiş bir simetrik blok şifreleme algoritmasıdır. Blowfish aşağıdaki özelliklere sahip olacak şekilde tasarlanmıştır;

Hız : Blowfish 32 bitlik mikro işlemciler üzerinde, veriyi bit başına 18 saat saykıl (clock cycle) hızıyla şifreler.
Sıkıştırılma : Blowfish, 5K'dan küçük belleklerde çalışabilmektedir.
Basitlik : Blowfish'in basit yapısının uygulanması kolaydır ve algoritmanın gücünün belirlenmesini kolaylaştırır.
Değişken güvenlik : Anahtar uzunluğu değişkendir ve 448 bit uzunluğunda olabilir. Bu durum daha yüksek hız ve daha yüksek güvenlik arasında denge sağlar.

Blowfish, 64 bitlik orijinal-metin bloklarını 64 bitlik şifreli metin blokları halinde şifreler. Blowfish, sayısız ürüne uyarlanmış ve çok sayıda tetkike maruz kalmıştır.

Blowfish şifreleme operasyonları aşağıdaki 2 basit adımdan oluşur;
Addition : Word'lerin toplama alınması, bu operasyon "+" ile gösterilir, mod 2^{32} işlemi icra edilir.
Bitwise-exclusive-OR : Bit-by-bit-exclusive-OR operasyonu, bu operasyon "⊕" ile gösterilir.

4.3. Açık Anahtar (Asimetrik) Şifreleme Yöntemleri

Açık anahtarlı şifreleme, simetrik şifreleme algoritmalarından radikal bir farklılık göstermektedir. Bu şifreleme yöntemi iki ayrı anahtar kullanan yöntemdir. İki anahtar kullanımının; güvenirlik, anahtar dağıtımı ve onaylama alanlarında önemli sonuçları vardır. Bu kriptografi yapısında, açık ve gizli anahtar olarak adlandırılmış olan bir anahtar çifti kullanılmaktadır.

Asimetrik algoritmalar da denilen açık anahtarlı algoritmalarda şifreleme için kullanılan anahtar ile şifre çözme için kullanılan anahtar birbirinden farklıdır. Anahtar çiftlerini üreten algoritmaların matematiksel özelliklerinden dolayı açık-gizli anahtar çiftleri her kişi için farklıdır, diğer bir deyişle her kullanıcının açık-gizli anahtar çifti yalnızca o kullanıcıya özeldir. Ayrıca şifre çözüm anahtarı, şifre anahtarından hesaplanamaz (en azından makul bir zaman dilimi içerisinde). Bu algoritmalara açık anahtarlı algoritmalar adı verilmesinin sebebi şifre anahtarı halka (kamuya/genel kullanıma) açılabilir olmasıdır. Bir yabancı bir iletiyi şifrelemek için şifreleme anahtarını kullanabilir, ancak sadece ilgili şifre çözüm anahtarına sahip bir kişi iletinin şifresini çözebilir. Bu sistemde, şifre anahtarına genellikle açık anahtar denir, şifre çözüm anahtarı da genellikle gizli anahtar olarak adlandırılır. Gizli anahtar kimi zaman özel anahtar olarak da adlandırılır, ancak simetrik algoritmalarla karışmaması için bu terim genelde kullanılmaz.

4.3.1. Asimetrik şifrelemede bilgi güvenliği

Elektronik iletişim, günümüzde kağıt üzerinde yazı yazarak yapılan her türlü iletişimin yerine geçmeye başlamıştır. Kişi/kuruluş/toplumlar, özel/kamusal/resmi haberleşmelerini kağıt üzerinden değil de elektronik iletişim ağları üzerinden yapabilmeleri için açık ağlar üzerinden iletilen bilginin güvenli iletilmesini isterler. Açık ağlarda bilgiyi tehdit eden unsurlar üç başlık altında toplanır:

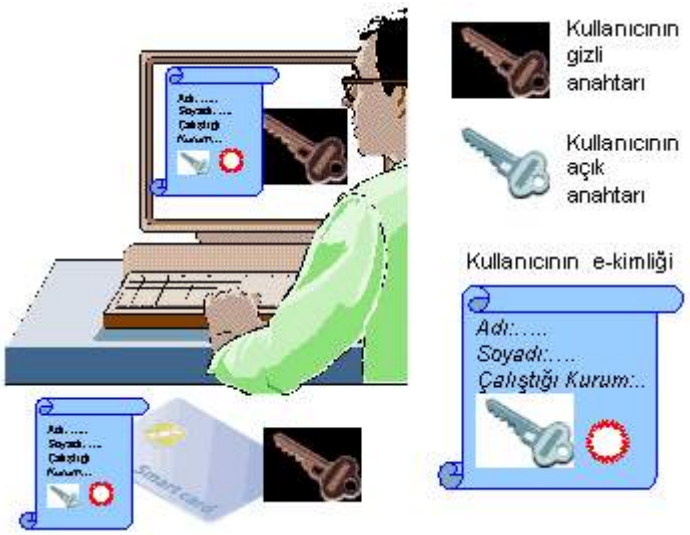
1. Başkası dinleyebilir
2. Bilgi değiştirilebilir
3. Kimlik taklit edilebilir

Bilgi güvenliği bu tehditlerin ortadan kaldırılması ile sağlanır. Bunun için kriptografide önerilen yöntemler çift (açık) anahtarlı bilgi güvenliği yöntemleridir. Tipik kriptografi uygulamaları güvenli iletişim, kimlik tanımlama, kimlik doğrulamadır. Bu uygulamalar sayısal şifreleme ve sayısal imza ana başlıkları altında aşağıda özetlenmiştir. Daha kompleks uygulamalar elektronik ticaret, elektronik kimlik belgeleme, güvenli e-posta, yeniden anahtar bulma ve bilgisayarlara güvenli erişim hakkı elde etme için geliştirilmiş sistemleri içerir.

Bir kullanıcının açık anahtarıyla şifrelenen bir mesajı yalnız ve ancak ona ait gizli anahtar çözebilir. Aynı şekilde, herhangi bir kullanıcının gizli anahtarıyla attığı sayısal imzanın doğrulanabilmesi, yalnızca onun açık anahtarını kullanarak mümkün olabilir.

Gizli anahtar, adından da anlaşıldığı gibi kişinin kendisi dışında kimsenin bilmemesi gereken ve saklı tutması zorunlu olan gizli bir bilgidir. Gizli anahtar genelde kişinin bilgisayarında anahtar kelimeyi sadece kendisinin bildiği şifreli bir dosya içinde saklanır. Açık anahtar kamuya açıktır, elektronik kimlik belgelerinin içinde diğer kişisel bilgilerle birlikte tutulur ve herkes birbirinin açık anahtarını e-kimliklerine ulaşmak suretiyle istediği zaman elde edebilir. Açık anahtarın içinde bulunduğu e-kimlikler kişinin bilgisayarında, bir kopyası da herkesin ulaşabildiği "sunucu"larda tutulur. (Şekil 4.7).

Açık anahtar bilgisiyle gizli anahtarın bulunabilmesi mümkün değildir. Kriptografide gizlilik, algoritmanın gizli tutulmasından değil, algoritmanın bazı matematiksel özelliklerinden dolayı şifrenin kırılmasının güçlüğünden ileri gelmektedir. Buradaki güçlük terimi, teknik olarak “hesaplanması pratik sınırlı sürelerde mümkün olmayan” anlamına gelir.



Şekil 4.7 Kullanıcı anahtar çiftlerinin bulunabileceği ortamlar

Şifreleme açık ağlardan gönderilen bilginin başkaları tarafından görülmesinin (dinlenmesinin) istenmediği, diğer bir deyişle yukarıda sözü edilen tehditlerin 1. şıkkında belirtilen durumun engellenmesi istendiği zaman yapılır. Bunun için çift anahtarlı bir kriptografik algoritma kullanılabilir. Buna göre, mesajı gönderen taraf:

Gönderilen bilginin sayısal içeriğini,
Mesajı alacak tarafın açık anahtarını,

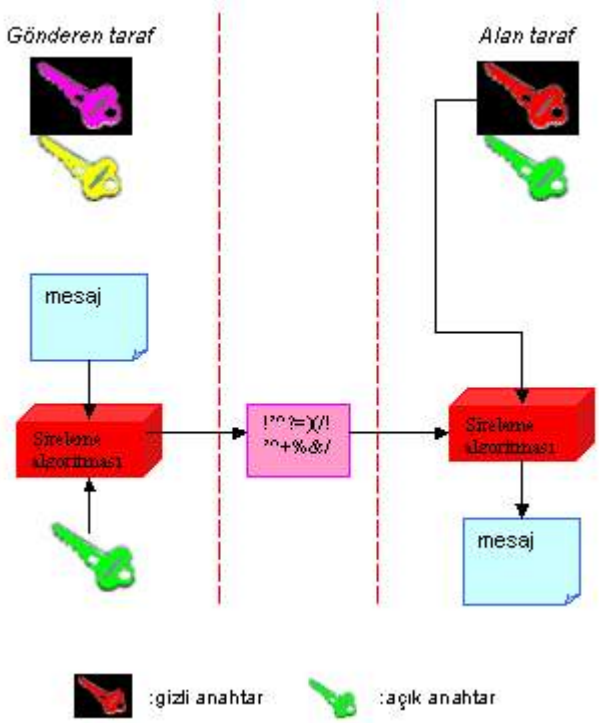
sayısal şifrelemede kullanır.

Mesajı alan taraf da, şifreli mesajı çözmek için şunlara gereksinim duyar:

- Şifreli mesajın sayısal içeriği
- Kendisinin gizli anahtarı.(Şekil 4.8)

Burada dikkat edilecek olursa, şifreli mesajın üçüncü taraflar tarafından dinlenebilmesi ancak "gizli anahtara" sahip olmaları ya da şifreli mesajı matematiksel yollarla deşifre etmeye çalışmaları ile mümkün olabilir. "Güvenlik

açısından iyi bir şifreleme" algoritması, gizli anahtar olmadan şifreli mesajı deşifre etmeye imkan tanımayan bir algoritmadır.



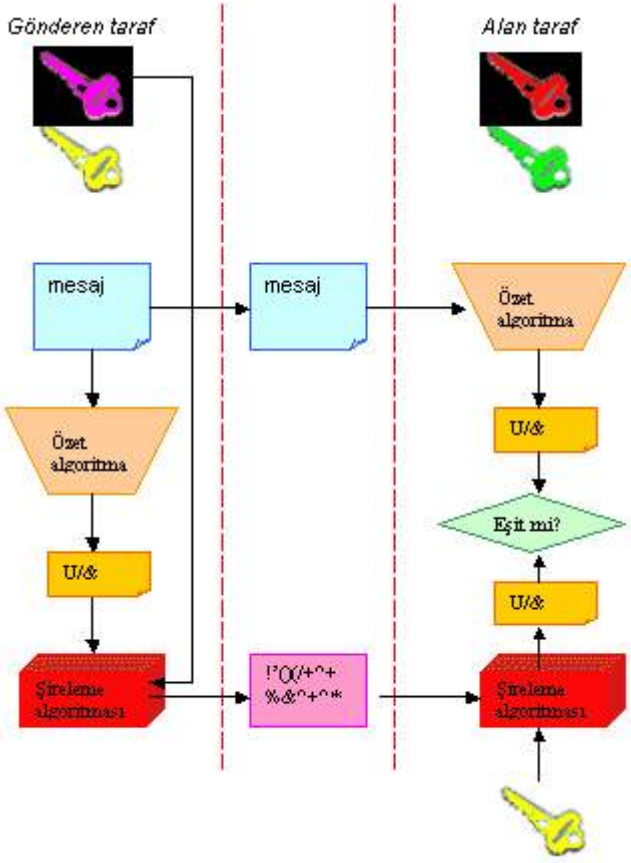
Şekil 4.8 Şifreli mesaj gönderilmesi ve alınması

Sayısal imza elektronik mesaja eklenmiş bilgidir. Çift anahtarlı bir kriptografik algoritmayla hazırlanan sayısal imza, hem gönderilen bilginin sayısal içeriğinin değiştirilmediğinin hem de gönderen tarafın kimliğinin ispatlanması, diğer bir deyişle yukarıda bahsedilen 2. ve 3. şıktaki tehditlerin engellenmesi için kullanılır ve şunlara bağlı olarak oluşturulur:

- Gönderilecek mesajdan üretilen "mesaj özetinin" sayısal içeriği,
- Gönderen tarafın kendi gizli anahtarı.

Sayısal imzanın doğruluğunu kanıtlamak için mesajı alan taraf şunları kullanır:

- Kendisine gelen mesajın ve sayısal imzanın sayısal içeriği,
- Gönderen tarafın açık anahtarı. (Şekil 4.9)



Şekil 4.9 Sayısal imzalı mesaj gönderilmesi ve alınması

"Mesaj özeti", gönderilecek mesajdan matematiksel yollarla üretilen sabit uzunlukta sayısal bilgidir. Bu işlem "hash" fonksiyonu olarak bilinir. Hash fonksiyonu bir kaç özelliği sağlar:

- Mesaj özeti anlamsız bir bilgidir.
- Hash fonksiyonu geri dönüşümü olmayan bir fonksiyondur. Diğer bir deyişle, herhangi bir mesajın özetine bakarak mesajın kendisini elde etmek mümkün değildir.
- Aynı özeti veren herhangi iki farklı mesaj bulmak da mümkün değildir.

Böylelikle, her mesajın farklı bir özeti olması ve dolayısıyla mesajda yapılacak en ufak bir değişikliğin imzayı geçersiz kılması sağlanmış olur. Sayısal imzalamada son adım, mesaj özetinin gönderen tarafın gizli anahtarıyla şifrelenmesidir. Sayısal imza mesaja eklenir ve mesaj ile birlikte alıcıya gönderilir.

Alıcının imzanın geçerliliğini kontrol etmesi iki adımda gerçekleşir. Alıcı sayısal imzayı karşı tarafın açık anahtarı ile çözerek varsayılan mesaj özetini elde eder. Diğer yanda mesajın tekrar özetini çıkarır. Son olarak bu iki özeti karşılaştırır. Bu özetlerin tıpatıp aynı olması, imzanın doğruluğunu gösterir.(Şekil 4.9)

Gizli anahtar, adından da anlaşıldığı gibi kişinin kendisi dışında kimsenin bilmemesi gereken ve saklı tutması zorunlu olan gizli bir bilgidir. Açık anahtar kamuya açıktır, elektronik kimlik belgelerinin içinde diğer kişisel bilgilerle birlikte tutulur ve herkes birbirinin açık anahtarını e-kimliklerine ulaşmak suretiyle istediği zaman elde edebilir.

Bazen iletiler gizli anahtar ile şifrelenip açık anahtar ile çözülür; bu elektronik imzalarda kullanılır.

4.3.2. Asimetrik şifreleme yönteminin genel prensipleri

Açık anahtar kriptografi simetrik kriptografi kullanımından daha yavaş olabilir. Bunun sebebi açık anahtar kriptografi`de kullanılan anahtarların daha uzun olması gerektiğinden ve algoritmanın daha çok işlemci gücüne ihtiyaç duymasındandır.

Genel-anahtarlı şifreleme sürecindeki gerekli adımlar şunlardır:

- Ağdaki her sunucu sistem bir çift anahtar yaratır, biri mesaj şifreleme diğeri de de-şifreleme için kullanılır.
- Her sistem şifreleme anahtarını genel bir listeye ya da bir dosyaya yerleştirerek yayınlar. Bu bir genel-anahtardır. Ortak kullanılacak anahtar gizli tutulur.

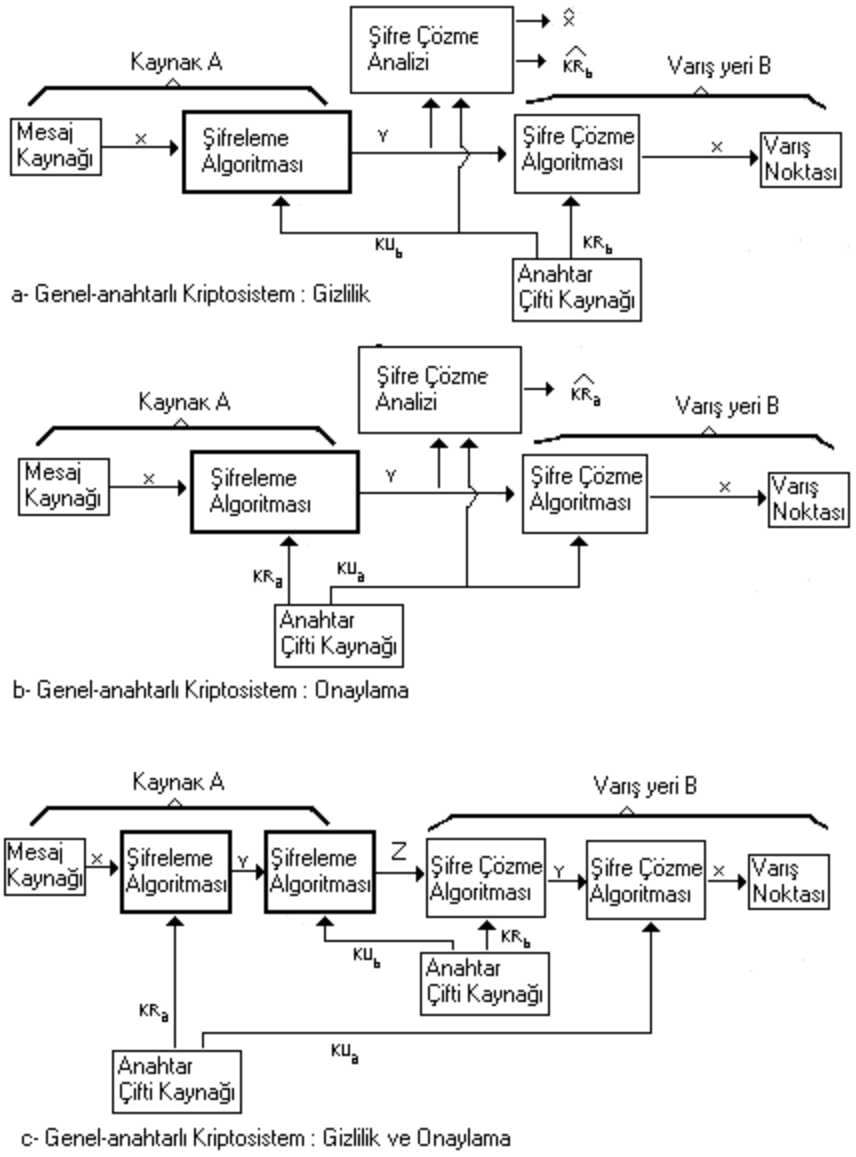
Eğer A, B'ye mesaj yollamak istiyorsa; A mesajı B'nin genel anahtarını kullanarak şifreler.

B mesajı aldığında, B kendi anahtarıyla mesajı de-şifre eder. Başka hiçbir alıcı mesajı okuyamaz, çünkü sadece B kendi özel anahtarına sahiptir.

Bu yaklaşımla, bütün katılımcılar genel-anahtarı elde edebilirler ve özel anahtarlar lokal olarak üretilmişlerdir. Bu yüzden özel-anahtarlar hiçbir zaman dağıtılmamalıdırlar. Sistem özel-anahtarlarını kontrol ettiği sürece, gelen bilgiler güvendedir. Herhangi bir zamanda bir sistem, özel-anahtarını ve ortak kullanımdaki genel-anahtarını değiştirebilir.

Bu adımları basit bir örnekle açıklamak gerekirse:

1. Abidin, Ayşe'ye İnternet üzerinden güvenilir bir mesaj yollamak istiyor. Ayşe'nin mesajı çözmenin bir yolunu bulması, ve mesajın, başka birinden değil Abidin'den geldiğinden emin olmalıdır. İlk olarak, Abidin mesajı hash fonksiyonu denen algoritmadan geçirir. Bu işlem, *message digest* olarak bilinen bir numara ürünü sunar. Bu message digest bir çeşit "**dijital parmak izi**" olarak çalışacak ve Ayşe'nin, başka kimsenin mesajı ele geçirmediğinden emin olmasını sağlayacaktır.
2. Abidin, Ayşe'nin açık anahtarı ile mesajı şifreler ve bu şifreli mesajla birlikte message digest'i şahsi anahtarını kullanarak şifreler. Bu sayede sadece Abidin'in şahsi anahtarının oluşturabileceği bir dijital imza ortaya çıkar.
3. Ayşe, Abidin'den gelen mesajı çözmek için Abidinin açık anahtarını kullanır. Ortaya Ayşe'nin açık anahtarı ile şifrelenmiş metin ile message digest çıkar. Mesaj Abidin'den gelmemiş olsaydı gelen mesaj çözülemezdi.
4. Ayşe, şahsi anahtarını kullanarak şifreli metini çözer ve ortaya orijinal olduğu var sayılan metin çıkar. Bu metinin message digesti alınır ve gelen message digest ile karşılaştırılır. Farklı ise mesaj iletim aşamasında değişikliğe uğramıştır.



Şekil 4.10 Genel anahtarlı kripto sistem

Şekil 4.10'te genel anahtarlı şifreleme şemasının esas elemanları gösterilmiştir. Bu elemanlar:

Orijinal-mesaj:X=[X1,X2,.....,Xm]

M :X'in(orijinal-mesajın)alfabedeki sınırlı harfleri

X^ :X'in(orijinal-mesajın)tahmini yaklaşımı

Şifreli-metin :Y=[Y1,Y2,.....,Yn]

N :Y'nin(şifreli-metnin)elemanları

KU_b :B'nin genel-anahtarı. Herkes tarafından ulaşılabilir.

KR_b :B'nin özel-anahtarı. Sadece B tarafından bilinir.

E :Şifreleme algoritması

D :De-şifreleme algoritması

K :Şifreleme/de-şifreleme anahtarı

K^ :K(anahtar) tahmini yaklaşımı.

Süreçler :

- A kaynağı bir mesaj için,X orijinal-metinden bir mesaj üretir.
- Mesaj, B adresine ulaşacak biçimde tasarlanmıştır.
- B, bir çift ilişkili anahtar yaratır. Biri genel-anahtar(KU_b), diğeri özel anahtar(KR_b)
- Özel-anahtar sadece B tarafından bilinir, genel-anahtar genele açıktır ve dolayısıyla A, genel-anahtara ulaşılabilir.
- X mesajı ve şifreleme anahtarı (KU_b),prosese girdi olarak alınır.

  A; şifreli-mesaj(Y)formunu oluşturur.

$$Y= E_{KUb}(X)$$

- Alıcı, bu mesaj için geçerli özel-anahtara sahipse,şifreli-mesajın(Y) ters dönüşümünü yaparak orijinal-mesajı(X)elde edebilir.

$$X=D_{KRb}(Y)$$

Bu süreçler devam ederken yapılabilecek kripto-analiz :

- Araya giren kişi, şifreli-metini (Y) fark ediyorsa ve B'nin genel-anahtarına (KU_b) ulaşabiliyorsa, fakat B'nin özel-anahtarına(KR_b) veya orijinal-metine(X) ulaşamıyorsa, KR_b ve/veya X 'i elde etmeye çalışır.
- Araya giren kişi, şifreleme (E) ve de-şifreleme(D) algoritma bilgisine sahip olduğu düşünülür.
- Eğer araya giren kişi sadece belirli bir mesajla ilgileniyorsa, X'i elde etmeye çaba gösterir. Bunu X olabilecek tahmini bir yaklaşımla (X^)'i oluşturarak yapar.
- Ama genellikle, araya giren kişi mesaj trafiğindeki sürekli akan mesajları okumak isteyecektir. Bu durumda tahmini yaklaşımla KR_b'den KR_b^'yi oluşturmaya teşebbüs etmelidir.

Şekil 4.10-a genel-anahtarlı şifrelemede güvenirliği gösterirken, şekil 4.10-b genel-anahtarlı şifrelemede onaylamayı gösterir ;

$$Y=E_{KRa}(X)$$

$$X=D_{KUa}(Y)$$

Bu durumda,

- A, B için bir mesaj hazırlar ve bunu B'ye göndermeden önce kendi özel anahtarını(KR_a) kullanarak şifreler.
- B, mesajı A'nın genel-anahtarını(KUa) kullanarak de-şifre edebilir.
- Çünkü, mesaj A'nın özel-anahtarı kullanılarak şifrelendiği için sadece A mesajı hazırlayabilir.
- Bu yüzden tüm şifrelenmiş mesaj, bir dijital imza işlevi görür.
- Buna ek olarak, A'nın özel-anahtarına ulaşmaksızın, bu mesajı değiştirmek mümkün değildir.
- Sonuç olarak,hem kaynak hem de veri bütünlüğü anlamında mesajın doğruluğunu kanıtlamış olur.

Bir dokümanın asıl yapısı olan bitlerin küçük bloklar halinde şifrelenmesi, sonuçları daha verimli kılar. Onaylayan diye tanımlanan böyle bir bloğun en önemli özelliği, onaylamayı yapanı değiştirmeden dokümanın da değiştirilmeyeceğidir. Eğer onaylayıcı, mesajı gönderenin özel-anahtarıyla şifrelendirilirse, mesajın kaynağı, içeriği ve sıralamayı güvence altına almak gibi bir servis de sağlar.

Bu şifreleme işlemi güvenliği sağlamaz, başka bir deyişle mesaj dinlenebilir fakat değiştirilemez. Tam şifrelemede bile şekil 4.10-b'de gösterildiği gibi güvenliğe karşı bir koruma yoktur. Çünkü herhangi bir gözlemci gönderenin genel-anahtarını kullanarak mesajı de-şifre edebilir.

Bununla birlikte Şekil 4.10-c'de gösterildiği gibi, genel-anahtarın iki kez kullanımıyla hem onaylama hem de güvenlik sağlanabilir.

$$Z=E_{KUb}[E_{KRa}(X)]$$
$$X=D_{KUa}[D_{KRb}(Z)]$$

Bu durumda işleme, mesajı gönderenin özel-anahtarını kullanarak mesajı şifreleme ile başlanır, bu da dijital imzayı sağlar. Sonra, mesajı alanın genel-anahtarını kullanarak mesaj tekrar şifrelenir. En son elde edilen şifreli-metin, uygun özel anahtarı olan alıcı tarafından de-şifre edilebilir. Böylece güvenirlik sağlanır. Bu yaklaşımın bir dezavantajı, her iletişimde zaten kompleks olan genel-anahtarlı şifreleme algoritmasını iki kez yerine dört kez kullanımıdır.

4.3.3. Asimetrik şifreleme yönteminin uygulamaları

Whitfield Diffie ve Martin Hellman açık anahtar kripto sistemler çağını başlatan bir anahtar değiş-tokuşu protokolü oluşturmak amacıyla sayı teorisini kullanmışlardır. Bundan kısa bir süre sonra, Ron **R**ivest, Adi **S**hamir ve Leonard **A**dleman şifreleme ve dijital imzalar kullanma kapasitesine sahip ilk gerçek kripto sistemi geliştirdiler. Sonraları farklı fikirleri kullanan pek çok açık kripto sistemler geliştirilmiştir. Bunların pek çoğunun güvensiz olduğu ispatlanmıştır. Ancak, Diffie-Hellman protokolü ve RSA şimdiye kadar kalanların en güçlüleri olarak görülmektedir. RSA pek çok uygulamada kullanılan bir algoritmadır. Mesajları şifrelemek için kullanılabileceği gibi dijital imzalarda da kullanılabilir.

Asimetrik şifreleme yöntemlerinin karışıklığa yol açan bir yönü vardır. Bu yöntemler iki anahtar kullanımı ile karakterize edilir; biri özel anahtar, diğeri genel kullanıma açık olan genel-anahtardır. Bazı kriptografik işlemler uygulamaya bağlı olarak, ya mesajı gönderenin özel-anahtarı veya alıcının genel anahtarıyla da her ikisi birden kullanılarak yapılır. Genel-anahtarlı kripto sistemleri yaygın olarak üç kategoride sınıflandırılabilir ;

Şifreleme/de-şifreleme : Mesajı gönderen mesajı, alıcının genel-anahtarı ile şifreler.

Dijital imza : Gönderen mesajı, özel anahtarı ile imzalar. İmzalama, mesaj ya da mesajın küçük bir bloğuna uygulanan bir kriptografik algoritma ile elde edilir.

Anahtar değiştirme : İki taraf, oturum anahtarını değiştirmek için yardımlaşır. İki tarafın özel-anahtarlarını kapsayan birçok değişik yaklaşım mümkündür.

Herhangi bir açık anahtar kripto sisteminin temel içeriği zor bir hesaplama problemidir. Kripto sistemin güvenliği, gizli anahtarın açık anahtardan elde edilebilmesi için bu zor problemin çözülmesi gerektiği ilkesine dayanmaktadır. Açık anahtar kriptografisinde kullanılan terminoloji şu şekildedir:

Asal Sayılar: Bir asal sayı 1 ve kendisi haricindeki hiçbir sayıya tam bölünemeyen bir sayıdır. Böylece, 2,3,5,7,11,... vs. gibi sayılar asal sayılardır. Sonsuz adet asal sayı vardır ve şu ana kadar bilinen en büyük asal sayı $(2^{6,972,593})$-1 dır.

Çarpanlara Ayırma: Her tamsayı, asal sayıların çarpımı ile temsil edilebilir. Örneğin, 10 = 2 * 5. Çarpanlara ayırma sanatı neredeyse matematiğin kendisi kadar eskidir. Çarpanlara ayırmayı hızlı yapan algoritmalar girdi sayısını kalan sayı asal oluncaya kadar küçük asal sayılar ile arka arkaya bölmeye çalışırlar. Bu ancak 10^{16} civarındaki tamsayılar için yeterli bir yoldur ve algoritma 10^8 e kadarki tüm asalları denemesi gerekir.

4.3.3.1. RSA algoritması

1978`de bulunan RSA, yaratıcılarının baş harflerini almıştır: Ronald **R**ivest, Adi **S**hamir ve Leonard **A**dleman. RSA günümüzde en çok kullanılan açık anahtar algoritmasıdır. Ayrıca RSA en çok test edilen algoritmalardan biridir. RSA hem bilgi şifrelemede hem de dijital imza sistemlerinde kullanılabilir. Kullanılan anahtarlar yeteri kadar uzun olduğunda güvenli olduğu düşünülür. RSA sistemindeki anahtarlar kullanılan belirli uygulamaya göre herhangi bir uzunlukta olabilir.

1976 yılında Stanford Üniversitesinden Diffie ve Hellman adlı iki araştırmacı farklı bir şifreleme sistemi önerdiler, bu sistemde bir tane şifreleme için(encryption

key) ve bundan farklı olarak bir tanede şifre çözmek için(decryption key) anahtar bulunmaktaydı ve decryption key encryption key den elde edilememekteydi.

Önerdikleri sistem şu şekilde idi:

Şifreleme algoritması kullanılan (anahtar) E ve, şifre çözme algoritması(anahtar) D aşağıdaki 3 ihtiyacı karşılamalı idi:

1. D(E(P))=P;(P=Orijinal mesaj)
2. E den D yi oluşturmak oldukça zor olmalı.
3. Şifrelenecek bilgiye yapılan saldırılardan dolayı E bozulmamalı.

Bu metodun çalışması şu şekildedir. Mesela, herhangi bir A kişisi özel bir mesaj yollamak istesin, öncelikle Ea ve Da algoritmalarını oluşturmalı ve yukarıdaki ihtiyaçları karşılamalı. Şifreleme algoritması ve anahtarı, Ea, herkes tarafından bilinmelidir. Bundan dolayı bunun adı public-key cryptography (genel anahtar şifrelemesi) olmuştur. Bu bilgi isteyenlerin okuyabileceği bir dosyaya konmalıdır. Ama şifre çözme anahtarı ise özel tutulmalıdır.

Şimdi A kişisi ile B kişisi arasında, önceden bir bağlantı kurmadan bir güvenlik kanalının nasıl oluşturulacağına bakılırsa: A'nın ve B'nin genel anahtarlarının herkes tarafından okunabilen bir dosyada olduğunu kabul edelim. A kişisi kendi mesajı olan P'ye Eb(P) işlemini uygular ve B'ye yollar. B ise kendi özel anahtarını(Db) kullanarak gelen şifreli mesajı çözer [Db(Eb(P))=P]. Kimse şifreli mesajı [Eb(p)] çözemez çünkü herkese ait bilgilerden Db yi elde etmek çok zordur. A ve B şimdi güvenli bir şekilde iletişim yapabilirler.

Genel anahtar şifreleme yöntemi bir yöne doğru rahatlıkla hesaplanabilen; fakat geriye doğru hesaplanması çok zor olan basit matematik problemlerine dayanıyor. Örnek olarak 5+3=8 ‘in hesaplanması basit bir işlemken, 8’in hangi sayılar toplanarak bulunduğunu hesaplamak bu kadar basit olmuyor. Hele söz konusu

olan büyük karmaşık sayılar olunca, bu hesabın geriye doğru yapılması neredeyse imkansızlaşıyor.

RSA çarpanlara ayırma ilkesini kullanmaktadır. Bu algoritma, genel anahtarlı şifreleme ve dijital imza işlemlerinde güvenli bir şekilde kullanılır. Bu algoritma aynı zamanda Netscape Navigator ve Microsoft Internet Explorer web browser programlarının uygulamaları olan Secure Socket Layer(SSL) ve kredi kartı işlemleri için Secure Electronic Transaction (SET) protokolü ile Mastercard ve VISA uygulamalarında kullanılmaktadır. Ayrıca S/MIME, PEM, MOSS ve PGP gibi gizli haberleşme protokolleri temel olarak RSA kullanır.

RSA algoritması açık anahtar şifreleme yönteminin temel bir uygulamasıdır. Bu şifreleme yönteminin diğer bir iyi tarafı ise önceden aralarında hiçbir görüşme yapmamış olan alıcı ve vericinin kendi aralarındaki iletişimin güvenli bir ortamda (güvenli gönderim ve mesajların doğrulanması) yapılmasıdır. İnternet'in şu anki durumunda iletişim güvenliği tam olmayan kanallar üzerinden işler. Bu durumda açık anahtar tekniğiyle internet'te güvenlik sağlanmıştır. Çünkü iki büyük asal sayı çarpıldığında ortaya çıkan sonucun, hangi iki asal sayının çarpılmasından elde edildiğini bulmak yine neredeyse imkansız. Genel anahtar şifrelemesinin diğer bir iyi tarafı ise, alıcının yollayıcının kimliğini tespit edebilmesi. Yollayıcı karşı tarafa mesaj göndermek istediğinde öncelikle kendi özel anahtarını kullanarak mesajı şifreliyor. Sonra alıcını genel anahtarını kullanarak mesajı tekrar şifreliyor. Alıcı ise mesajı aldıktan sonra işlemleri geriye doğru tekrarlıyor; önce kendi özel anahtarını kullanarak mesajı çözüyor ve bundan sonra yollayıcının genel anahtarını kullanarak mesajı ikinci kez çözüyor. Eğer mesaj anlaşılabilen bir formatta ise , bu mesajın düşünülen kişiden geldiğini doğruluyor.

Bütün bu özelliklere rağmen algoritma mükemmel değildir. Çok karmaşık matematik işlemleri gerektiren bu algoritma uzun mesajları şifrelemekte yavaş kalıyorlar.

Şifre kırma tekniklerinin gelişmesine paralel olarak şifreleme yöntemleri de sürekli gelişim gösteriyor. Şifreleme yöntemine karşı en önemli tehdit, araya girebilecek "üçüncü şahıs" tehlikesi. Bu genel anahtar şifrelemesinin en zayıf yönünü oluşturuyor.

Yollayıcı bir mesaj göndermek istediğinde , üçüncü bir kişinin alıcının kimliğini taklit ettiğini fark edemeyebiliyor. Üçüncü kişinin mesajı yollayan kişiyi bu şekilde kandırması halinde, yollayıcı farkında olmadan üçüncü kişinin genel anahtarı ile şifrelemeyi yapmış oluyor. Bu durumda üçüncü kişi, şifreli mesajı çözebilme olanağını elde ediyor. Bunu engellemenin tek yolu, yollayıcının bir şekilde , genel anahtarın gerçekten mesajın ulaşmasını istediği kişiye ait olduğunu anlaması. Bunu sağlamanın yollarından birisi de, güvenilir bir otoritenin bu anahtarları onaydan geçirmesi ve imzalaması.

RSA Algoritmasının içeriği ve çalışması aşağıdaki gibidir:

Gizli (private) ve açık (public) key çifti üretilir.

P ve Q şeklinde çok büyük (mesela, 1024 bit) iki tane birbirinden farklı asal sayı bulunur.

N=P*Q ve Z=(P-1)* (Q-1) hesaplanır.

Z ile ortak böleni 1 olacak şekilde bir E sayısı bulunur.

Public key {E,N} olarak belirlenir.

$D=E^{-1}$ mod Z olacak şekilde bir D sayısı bulunur.

Private key [D,N] olarak belirlenir.

Şifrelenecek mesajı m kabul edersek bu mesaj binary olarak $2^k<N$ olacak şekilde k bitlik kısımlara ayrılır [m=m(1)+m(2)+m(3)+...+m(n)] . Daha sonra encryption için her bir kısma $C(i)=m(i)^E$ mod N işlemi uygulanır.

Private key [D,N] kullanılarak şifre çözülür. [$m(i)=C(i)^D$ mod N]

RSA hesaplaması, iki gizli büyük *p, q* asalı için, tamsayı modulosu $n = p * q$ ile yapılmaktadır. *m* mesajını şifrelemek için, *m*'nin küçük bir açık üs (public exponent) ile kuvveti alınır. Deşifre etmek için, $c = m^e \pmod n$ şifreli-mesajının alıcısı $d = e^{-1} \pmod{(p-1)*(q-1)}$ çarpılabilir tersini hesaplar ve $c^d = m^{e*d} = m \pmod n$ yi elde eder. Gizli anahtar *n, d*'den oluşur (öyle ki *p* ve *q* unutulabilir); açık anahtar sadece *n, e*' yi içermektedir.

Anahtar genişliği (modülüsün genişliği) makul bir güvenlik düzeyi için *1024* bitten daha geniş olmalıdır.

Asimetrik kriptografi`de bir özel anahtar, onu eşleyen açık anahtarın tersidir. Bir açık anahtar algoritması sadece eğer bir kişinin açık anahtarından özel anahtarı (uzun bir süre içinde) bulunamıyorsa güvenli sayılır. RSA algoritmasında özel anahtarın açık anahtardan bulunamamasının sebebi büyük sayıların asal çarpanlarını bulmadaki zorluktandır. RSA algoritmasındaki diğer önemli bir kavram, aralarında asal çarpanlar kavramıdır (relatively prime factors). Eğer iki sayı birbirlerinin çarpanları değilse bu iki rakama `aralarında asal` denilir. Örneğin, 3 ve 7 rakamları aralarında asaldırlar çünkü 3, 7`nin bir çarpanı değildir. Fakat, 3 ve 6 aralarında asal değildirler, zira 3, 6`nin bir çarpanıdır (6= 3*2).

RSA algoritması modüler matematik kullanır. Yani, sadece sıfırdan başlayan ve belirli bir sayıya kadar (o sayı dahil değil) olan sayılar (modulus) kullanılır. Eğer mod olarak 20 kullanılıyorsa denklik bağıntısı gereği 20 sayısı yerine 0 (20 mod 20) kullanılır, 21 yerine de 1 (21 mod 20) kullanılır. Benzer şekilde 39 sayısı 19 (39 mod 20) , 40 da 0 (40 mod 20) olarak hesaplanır.

Bu tip bir denklem kullanarak bir rakamın modüler terimlerle nasıl ifade edilebileceği bulunabilir. Matematikte iki sayının çarpımının belli bir fonksiyon sonucu 1 sayısını üretiyorsa o iki rakam birbirinin tersi kabul edilir. Modüler matematikte de aynı kural geçerlidir. Örneğin, eğer modulus 20 ise, 3 ve 7 rakamları birbirinin tersidir. Bunun sebebi, 3*7=21'dir ve 21 mod 20= 1 olarak gösterilir. Kullanılan modulus yaratılan her bir anahtar seti için farklı olabilir. Bir modulus

seçmek için rast gele iki büyük asal sayı seçilir (p ve q). İkisini birbiriyle çarpımından çıkan sonuç, n, şifreleme ve deşifreleme için kullanılacak modulus değeridir. Üslü ifade (exponent), E, olarak bilinen, n`den küçük değer açık anahtar değeri olarak hesaplanır. Bu sayının asal olması gerekmiyor fakat tek sayı olmalı. Açık anahtar değeri E (p-1)(q-1)'e ilişkili asal olmak zorundadır. Bir mesajı (m) şifreli hale (c) çevirmek için bu E değeri kullanılır. Elde edilen şifreli metni çözebilmek için açık anahtarın tersi olan değer bilinmelidir. Bu değer özel anahtardır (D). Bu D değeri ile şifreli metin çözülebilir. D'yi bulmak için modulus n yi değil farklı bir modulus kullanılır, (p-1)(q-1). Özel anahtar açık anahtarın tersi olduğundan bu denklemde bir sorun yaşanmaz. Açık anahtar E ve özel anahtar D arasındaki ters ilişki RSA algoritmasının şifreli yazıyı Orijinal mesaja başarılı bir şekilde çevirmesini sağlar. Eğer özel anahtar D yi bilinmiyorsa açık anahtar E ile şifrelenmiş şifreli metin çözülemez. Fakat açık anahtar E biliniyorsa ve ayrıca modulus n in asal çarpanları p ve q da biliniyorsa bu denklemi kullanarak özel anahtar kolayca bulunabilir. Bu yüzden p ve q yu gizli tutmak önemlidir. Eğer sadece açık anahtar E ve modulus n biliniyorsa özel anahtarı hesaplayabilmek için önce modulus un asal çarpanlarının bulunması gerekir. Fakat yeterli derecede büyük sayılar kullanıldığında asal çarpanların bulunması nerdeyse imkansızdır.

Mesajları açık anahtar algoritması ile şifrelemek ve açmak simetrik sistemden çok daha fazla işlem gücü harcar. En hızlı RSA çipi 512-bitlik asallar kullandığında saniyede 600 kbit çıktı verebilir. Benzer DES donanım uygulamaları 1000 ile 10000 kat daha hızlıdır. Ve DES yazılım uygulamaları RSA algoritmasından 100 kat daha hızlı şifreleme yapar.

Simetrik şifrelemeden yavaş olmasına rağmen açık anahtar şifreleme daha güvenlidir. Örneğin, eğer simetrik şifreleme kullanarak bir mesaj gönderilirse her alıcıya özel anahtarın gönderilmesi için güvenli bir yol bulunması gerekir. Simetrik şifreleme kullanımında haberleşme için kullanılacak anahtar konusunda özel olarak anlaşılması gerekir. Fakat açık anahtar şifreleme çok yönlüdür. Açık anahtarı bilinen herhangi birine, önceden bir anlaşma yapmadan gizli mesajlar gönderilebilir. Açık

anahtar sistemleri çoğu zaman özel anahtarın güvenli olarak gönderilmesi için dijital zarf (dijital envelope) yaratılmasında kullanılırlar.

Bir dijital zarf yaratmak için mesaj gizli anahtarla şifrelenip alıcıya gönderilir. Alıcı mesajı açmak için paylaşılan gizli anahtara ihtiyaç duyar ve ona gizli olarak göndermeniz gerekir. Paylaşılan gizli anahtarı alıcının açık anahtarı ile kriptolanarak açık ağ üzerinden gönderilebilir.

Dijital zarf zaman kazandırır çünkü mesajın kendisi değil sadece paylaşılan gizli anahtarın asimetrik şifreleme ile şifrelenip açılması gerekir.

Bugün kullanımda olan bütün açık anahtar kripto sistemleri, çok güçlü bilgisayarlar tarafından bile, hatırı sayılır bir süre içerisinde çözülmesi imkansız belirli matematik problemlerine dayanır. Bu tip problemlerden biri çok geniş bir sayının asal çarpanlarını bulmaktır. Bir sayının asal çarpanları sonuç olarak sayıyı veren asal sayılardır (Bir asal sayı sadece kendisine ve bire bölünebilen sayılardır). Örneğin 91`in asal çarpanları 7 ve 13`tür. Fakat 899 gibi bir sayının asal çarpanlarını bulmak daha zordur. Sayı büyüdükçe asal çarpanlarını hesaplamak ta zorlaşır (899`un asal çarpanları 29 ve 31 dir). RSA açık anahtar sistemine matematiksel olarak iki geniş asal sayıyı asal çarpanlarına ayırarak saldırılabilir. Eğer bu iki asal çarpanı bulunabilirse bu değerler açık ve özel anahtar olarak kullanılabilir. Anahtar yaratmada büyük sayılar kullanmak bu yöntemi zorlaştırır ve pratikte asal çarpanlarına ayırmayı imkansız hale getirir. Örneğin; teorik olarak 512-bitlik bir anahtarın asal çarpanlarını 1 MIPS (saniyede 1 milyon işlem) lik bir bilgisayarla bulmak için 420 bin yıl gerektiği hesaplanmıştır. Buna rağmen 512-bitlik anahtarlar potansiyel olarak güçsüz olarak düşünülmektedir. 1024-bitlik anahtarlar pek çok amaç için yeterli derecede güçlü olarak görülmektedir. Eğer gelecekte büyük sayıların asal çarpanlarını bulmak için yeni ve hızlı bir yöntem bulunursa RSA sistemi de güvensiz hale gelecektir. Fakat şu an RSA bu tip saldırılardan hiçbir şekilde etkilenmemektedir. Sebebi ise, anahtar için kullanılan 'modulus' boyutu gelişen teknolojiye ayak uyduracak şekilde arttırılabilir olmasıdır.

Anahtar uzunluğu seçilirken sadece güvenliği değil performansı da düşünülmelidir. Anahtar boyutu büyüdükçe bilgiyi şifreleme ve şifreyi çözme için ihtiyaç duyulan işlem gücüde artar. Bir açık anahtar sistemini kırmanın sistematik olmayan yollarından biri, mesajlarını okumak istediğiniz kişinin özel anahtarını ele geçirmektir. Örneğin, eğer özel anahtar kullanıcının bilgisayarı gibi belli bir yerde saklanıyorsa çalınması mümkündür. Ya da anahtarı bilen başka birisinden temin edilebilir.

4.3.3.1.1. RSA algoritmasının önemli özellikleri

RSA nın yasal kullanımı ile kolayca hesaplanabilenler;

- Birbirine uyan public/private anahtar çifti oluşturulabilir.
- Eğer public-key varsa ,şifreleme ve imzalama işlemleri yapılabilir.
- Eğer private-key varsa, şifre çözme ve imza onaylama işlemleri yapılabilir.

RSA nın yasalara aykırı kullanımı ile hesaplaması çok zor olanlar;

- public-key varken private-key in oluşturulması.
- private-key yokken şifreli mesajın çözülmesi.

RSA şifreleme algoritmasının güvenli olmasını şu şekilde açıklanabilir:

Bu algoritmaya zarar vermek için bilinen en etkin yol N'in asal çarpanları olan P ve Q nun bulunmasıdır. Ancak P ve Q çok büyük asal sayılar olduğundan ötürü bulunmaları neredeyse imkansızdır. Ayrıca public key {E,N} belli iken D yi, P ve Q olmadan bulmak çok zordur.

4.3.4. Geleneksel şifreleme ile genel anahtarlı şifreleme farkları

Tablo 4.4.'te geleneksel ve genel-anahtarlı şifrelemenin bazı önemli yönlerini özetleyen bir tablo verilmiştir. İkisini birbirinden ayırmak için, geleneksel şifrelemede kullanılan anahtarı gizli-anahtar(secret key) olarak, genel-anahtarlı şifrelemede kullanılan anahtarları ise, genel-anahtar(public key) ve özel-anahtar(private key) olarak adlandırılmıştır. Özel-anahtar daima gizli tutulur fakat geleneksel şifrelemede kullanılan anahtarla karıştırılmaması için gizli-anahtardan ziyade özel-anahtar olarak adlandırılır.

Tablo 4.1 Geleneksel şifreleme ve genel anahtarlı şifreleme karşılaştırması

Geleneksel Şifreleme	**Genel Anahtarlı Şifreleme**
Aynı algoritma ve aynı şifreleme anahtarı hem şifreleme hem de şifre çözmede kullanılır.	Şifreleme ve şifre çözmek için bir algoritma fakat şifreleme ve şifre çözme için farklı anahtarlar kullanılır
Gönderici ve alıcı aynı algoritmayı ve aynı anahtarı kullanır.	Gönderici alıcının açık anahtarını bilmelidir. Gönderici ile alıcının anahtar çiftleri birbirinden farklıdır.
Şifreleme için kullanılan anahtar gizli tutulmalı	İki anahtardan biri gizli tutulmalı diğeri erişime açık olmalıdır.
Algoritma bilgisi ve şifreli metin örnekleri anahtarı belirlemede yeterli olmamalı	Algoritma bilgisi,anahtarlardan birinin ve şifreli-metin örnekleri,diğer anahtarı belirlemede yeterli olmamalı

4.4. Şifrelemede Kullanılan Anahtar Boyutları

ABD, anahtar uzunluğu 40 bit'i geçmeyen şifreleme algoritmaları içeren yazılımların ihracını yasaklamıştır. 40 bitlik bir anahtar için $\mathbf{n=2^{40}}$ veya **n=1 099 511 627 776** (bir trilyon doksan dokuz milyar beş yüz on bir milyon altı yüz yirmi yedi

bin yedi yüz yetmiş altı). 1995'de yapılan bir yarışmada RC4 algoritması ile 40 bitlik bir anahtarla şifrelenmiş WWW üzerinden yapılan bir kredi kartı işlemi elinde sadece mütevazi bir bilgisayar laboratuarı olan bir öğrenci tarafından 3 buçuk saatte çözülmüştür.

Anahtarın deneme-yanılma yöntemiyle bulunmasını engellemek için, bugünkü süper bilgisayarlardan milyonlarca kat daha hızlı çalışan bir bilgisayarla bile milyarlarca yıl sürmesi için, yaklaşık 100 bitlik bir anahtar uzunluğunun uygun olduğu düşünülmektedir.

Tablo 4.2.'te farklı anahtar boyları için, saniyede bir milyon, bir milyar ve bir trilyon şifre deneyebilen bilgisayarlar için kırma zamanları verilmiştir.

Tablo 4.2 Farklı anahtar boylarında şifre kırma süreleri

Anahtar Uzunluğu n	**2^n**	**10^6 şifre/s hızında ortalama çözme süresi**	**10^9 şifre/s hızında**	**10^{12} şifre/s hızında**
32 bit	~$4x10^9$	36 dak	2.16 s	2.16 ms
40 bit	~10^{12}	6 gün	9 dak	1 s
56 bit	~$7.2x10^{16}$	1142 yıl	1 yıl 2 ay	10 saat
64 bit	$1.8x10^{19}$	292 000 yıl	292 yıl	3.5 ay
128 bit	$1.7x10^{38}$	$5.4x10^{24}$ yıl	$5.4x10^{21}$ yıl	$5.4x10^{18}$ yıl

Türkçe **şifre çözüm** diyebileceğimiz kriptanaliz, şifre bilimin (kriptolojinin) temel etkenlik alanıdır. Temel olarak, şifre çözümün amacı, kullanılan şifrenin zayıflıklarından ve şifrelenen metin hakkındaki bilgilerden yola çıkarak bütün anahtarları deneme zahmetinden kurtulmaktır. Kriptanalist, veya şifre çözücü düşmanın şifrelerini çözmekle uğraşan teknisyendir. Kriptolog ise şifre bilimcidir.

Nazilerin 1940-1944 savaşında kullandıkları ünlü Enigma şifreleme makinesi 116 bitlik bir anahtar uzunluğuna sahipti. Yani bugünün sağlam şifrelerinin kullandıkları boyda idi. Buna rağmen, savaşta şifre bilimcilik yapan bilgisayar

biliminin babası İngiliz Alan Turing, Enigma şifresinin yapısal zayıflıklarını kullanarak, Colossus isimli ilk tüplü bilgisayar yardımıyla da Nazilerin iletişimlerini çözmeyi başardı

Sezar şifresi basit bir alfabe değiştirme (*ing. substitution*) şifresidir. Alfabedeki harflerin yerlerini rasgele değiştirerek daha güçlü bir şifre yapılabilir. Bu durumda **n** alfabedeki harf sayısı ise, **n!** yani **28! ~= $3x10^{29}$** değişik şifre alfabesi olur. Yani deneme-yanılma yöntemiyle hangi alfabenin kullanıldığını bulmak hemen hemen olanaksızdır. Buna rağmen, "alfabe değiştirme şifresinin şifre çözümü, özellikle harf sayısı görece az olan Latin alfabelerinde çok basittir. Herhangi bir Türkçe, Fransızca veya İngilizce (veya Almanca veya İtalyanca) metin alıp hangi harfin ne kadar kullanıldığına bakarak, harf kullanım dağılımını analiz etmenin çok da faydası yoktur.

Alfabe değiştirme yöntemiyle şifrelenmiş bir metinde, yapılacak ilk iş metinin içerdiği bütün harflerin (veya şekillerin) kullanım istatistiğini hesaplamak olmalıdır. Ondan sonra, metnin hangi dilde yazıldığına dair tahminler yürütüp, o dilin harf kullanım sıklıklarını hesaplayıp, ikisini karşılaştırarak ve biraz tahmin yürüterek kolayca şifre çözülebilir.

Şifre çözüm konusundaki araştırmaların büyük çoğunluğu devletler tarafından yapılıyor. Örnek olarak ABD'nin şifre bilim üniversitesi denilebilecek NSA (National Security Agency veya Ulusal Güvenlik Kurumu) dünyanın en çok matematikçi istihdam eden kurumu. Devletler, matematikçilerinin şifre çözüm ve şifre tasarımı hakkında bulduklarını, üstünlüklerini koruyabilmek için saklı tutmaktadırlar (bu elbette bütün teknolojiler için geçerli). Dolayısıyla kamuda bilinen şifreleme algoritmalarının güvenilirliğini değerlendirmek için sivil şifre bilimcilerin sınırlı kaynaklarla yaptıkları çözme girişimlerinden başka bilgi kaynağı yoktur.

Şifre bilimciler bir algoritmanın güvenilirliği hakkında 10 yıldan daha uzun bir süre için tahmin yürütmenin epey zor olduğunu ifade ediyorlar.

4.4.1. Kriptografik sistemlerin güvenilirliği

İyi kriptografik sistemler davetsiz misafirlerin sistemi kırmasına mümkün olduğunca izin vermeyecek şekilde tasarlananlardır. Uygulamada kırılamayacak sistemler yapmak olasıdır ancak teoride bu ispatlanamaz. Teoride, herhangi bir anahtarlı kriptografik metot olası tüm anahtarların denenmesi ile kırılabilir. Eğer tüm anahtarların denendiği deneme-yanılma (brute force) kullanımı tek yolsa, gerekli hesaplama gücü anahtarın uzunluğu ile üstel olarak artar.

Olası anahtarların denenmesi işlemi, anahtarın uzunluğu arttıkça güçleşecektir. Örneğin 5 bitlik (5 adet 0 veya 1'den oluşan) bir anahtarın, en fazla 2^5=32 farklı anahtarı olabileceğinden, bu anahtarların sırayla denenmesi, şifrenin çözülmesine yeterli olacaktır. 32 bitlik bir anahtar 2^{32}(yani 4,294,967,296) adım alır. Bu herhangi bir ev bilgisayarı ile denenebilir. 40 bitlik anahtara sahip bir sistem için 2^40 adım gerekir ki, bu da birçok üniversitenin ve hatta bazı küçük şirketlerin bile sahip olabileceği bir güçle mümkündür. 56 bit anahtarlı bir sistem (DES gibi) esaslı bir zahmet gerektirir (çok sayıda ev bilgisayarının güç paylaşımı ile bunu kırmak birkaç ay alır), ama özel donanımlarla kolayca kırılabilir. Özel donanımların maliyetleri de doğal olarak yüksektir, ama organize suç örgütleri, büyük hükümet ve şirketler bunları alabilirler. Saniyede 3 trilyon anahtarı deneyebilen bir makine ile (yaklaşık 1 milyon dolar yatırımla elde edilebilecek bir makine), 56 bitlik DES algoritmasını 3,5 saat içinde kırılabilir. Aynı makine 80 bitlik anahtar kullanan bir algoritmayı 6.655 yılda, 128 bitlik anahtar kullanan bir algoritmayı ise 2.000.000.000.000.000.000 yılda kırabilir. Gelişen teknoloji ve işlemci hızları ile bu tip işlemlerin daha hızlı gerçekleştirilebileceği açıktır. 64 bitlik anahtarlarda şimdiden kırılabilir durumdadırlar. 80 bitli anahtarlar ve 128 bitli anahtarlar deneme-yanılma ile muhtemelen kırılamayacaklardır. Ancak anahtar uzunluğu tek önemli konu değildir. Pek çok şifreleme algoritması olası tüm anahtarlar denenmeden de kırılabilir.

Özgün şifreleme algoritmaları tasarlamak daha kolay olabilir, ancak gerçek uygulamalarda tavsiye edilmemektedir. Çoğunlukla, algoritmanın gizliliğine dayanan

algoritmalar güvenli değillerdir. Özellikle yazılım yoluyla, herhangi bir kimse algoritmayı tersine çevirip çalıştırabilecek (disassamble) bir yöntem bulabilir.

Kaotik sistem kullanarak elde edilen değerler başlangıç değerlerine hassas olarak bağlıdır. Başlangıç değerleri olan x,y ve z değerleri üzerinde yapılan küçük bir değişiklik ileriki aşamalarda tahmin edilemeyecek sonuçlar vermektedir. Lorenz Kaotik sistemin ürettiği sayılar rasgelelik testlerini başarıyla geçmiştir.[7]. Bu da Lorenz kaotik sisteminin tahmin edilemeyeceğini göstermektedir. Lorenz kaotik sistemin ürettiği değerler 3 tane olduğu için şifrenin kırılması için denenmesi gereken anahtar sayısını 3'e katlamaktadır. Örneğin değişken aralıklarının altı haneli olması durumunda 60 bitlik simetrik algoritmayla kırılma süresi denk düşmektedir. Bu saniyede 10^{12} işlem yapan bir bilgisayar 3.5 ay sürmektedir. Değişkenlerin alabileceği değer aralıkları 12 basamağa çıkartılırsa 128 bitlik bir simetrik algoritmayla kırılma süresi eşit olmaktadır. Bu da saniyede 10^{12} işlem yapan bir bilgisayarda $5.4x10^{18}$ yıl sürmektedir.

Açık-anahtarlı şifreleme yöntemlerinde kullanılan anahtarların uzunluğu simetrik şifreleme yöntemlerinde kullanılanlardan daha uzundur. Bunun nedeni, kriptanalistler için kullanılan ekstra yapıdır. Burada problem doğru anahtarın tahmin edilmesi değil, gizli anahtarın açık-anahtardan türetilmesidir. RSA da bu işlem iki asal çarpanı olan bir tamsayının üretilmesi ile yapılmaktadır. RSA kripto-sisteminin karmaşıklığı hakkında biraz bilgi vermek gerekirse, 256 bitlik bir modulus evde kolayca ve 512 bitlik anahtarlar üniversitedeki araştırma grupları tarafından birkaç ay içinde kırılabilir. 768 bitlik anahtarlar muhtemelen uzun vadede güvende sayılmazlar. 1024 ve daha büyük bitli anahtarlar RSA ya karşı büyük kriptografik ilerlemeler kaydedilmedikçe güvende sayılırlar.

Görüldüğü gibi, teorik olarak kırılabilir kabul edilen algoritmaların pratikte kırılması, anahtar büyüdükçe imkansızlaşmaktadır.

4.4.2. Şifre çözme algoritmalarının analizi

Şifrelemenin bütün esprisi düz metini (veya anahtarı, ya da her ikisini) davetsiz misafirlerden (istenmeyen kişilerden, düşmanlardan, saldırganlardan, rakiplerden) gizli tutmaktır.

Şifre analizi anahtarı kullanmadan bir iletinin düz metnini eski haline getirme bilimidir. Bu işlem orijinal metin veya şifreleme anahtarı veya her ikisini de bulmaya yönelik bir işlemdir. Burada kullanılan strateji doğal olarak şifrelemenin tasarımına ve şifre çözen analistin bilgisine bağlıdır. Analistin sahip olduğu bilgi ve teçhizat olası saldırıları belirler. Başarılı bir şifre analizi düz metini veya anahtarı eski, ilk haline getirir.

Bir kriptografik tasarımın güvenliğini etkileyen faktörler makinelerin hesaplama gücü, anahtarın rasgeleliği, kullanılan anahtar ve algoritmanın kendisidir. Bir algoritma hesaplama kaynakları ile kırılamaz ise hesaplanabilir olarak güvenlidir. Ancak, hesaplama gücü son yıllarda çok gelişmiştir. Ayrıca çoğu algoritma saldırıları, paralel bilgisayarlarda gerçekleştirilmektedir.

Algoritmanın performansını etkileyen diğer bir faktör de anahtar uzunluğudur. Saldırıları etkisiz kılmak için uzun anahtarlar kullanırız. Bir anahtar sıklıkla değiştirilmeli ve eksi anahtarlar yeniden kullanılmamalıdır.

Çoğu insan algoritmanın gizlenmesi ile tasarımı güvenli kılabileceklerini düşünür. Ama gerçekte, bütün bilgisayar programları tersine çevrilebilir. İyi bir algoritma endişe duyulmadan açıklanabilir. Şifre çözme anahtarı olmadan, şifreleyen kişi bile şifrelenmiş mesajı deşifre edemez.

Kullanılan şifreleme algoritması bilinmese bile genellikle saldırganın şifreleme algoritmasını bildiği kabul edilmelidir. Bu özelliğe dayanarak muhtemel saldırılardan biri, olası tüm anahtarları denemek olan ‘Brute force’ yaklaşımıdır. Bu yüzden saldırgan değişik istatistiksel uygulamalar, çeşitli testler yapmak zorundadır.

Şifreli bir metin orijinal metini belirleyebilecek yeterli bilgi bulundurmuyorsa, bu şifreleme algoritması şartsız olarak güvenlidir. Burada kişinin ne kadar zamanı olduğu önemsizdir, şifreli metinin çözülmesi imkansızdır.

Şifreleme algoritması seçilirken aşağıdaki kriterlerden biri yada her ikisine dikkat edilmelidir

- Şifreyi kırmanın maliyeti şifrelenmiş bilginin maliyetini aşmalı.
- Şifreyi kırmak için geçen zaman bilginin yaşam seyrindeki geçerlilik süresini aşmalı.

Eğer şifreleme yöntemi yukarıda sözü edilen iki kriteri karşılıyorsa bu yöntem hesaplama yönünden güvenlidir denir.

En önemli saldırı tekniklerinden bazıları aşağıda verilmiştir:

4.4.2.1. Sadece şifreli metin saldırısı

Bu saldırı tipinde, saldırıyı yapan kişi mesajın içeriği hakkında hiç bir şey bilmemektedir ve sadece şifreli-metni kullanarak çalışmalıdır. Uygulamada düz metine (plain text) (pek çok mesaj türü sabit başlık formatlarına olduğundan) ilişkin tahminler yapmak genelde olasıdır. Sıradan mektuplar ve belgeler bile kestirilebilir bir şekilde başlamaktadır. Örneğin, pek çok klasik saldırıda şifreli metinin frekans analizi kullanılmaktadır, ancak modern şifreleme yöntemlerine karşı bu yöntem iyi çalışmamaktadır.

4.4.2.2. Bilinen düz metin saldırısı

Saldırgan şifreli metinin bazı kısımlarından düz metni tahmin edebilir veya bölebilir. Geriye kalan iş bu bilgiyi kullanarak şifreli metin bloklarını çözmektir. Bu işlem, veriyi şifrelemek için kullanılan anahtarın belirlenmesi ile yapılabilir. En sık kullanılan Bilinen Düz Metin Saldırısı, blok şifrelemeye karşı lineer kriptanaliz saldırısıdır.

4.4.2.3. Seçilmiş düz metin saldırısı

Saldırgan bilinmeyen anahtar ile şifrelenmiş istediği her metni elde edebilmektedir. Buradaki iş, şifreleme için kullanılan anahtarı belirlemektir. Bazı kripto sistemler, özellikle RSA, seçilmiş-düz metin saldırılarına karşı açıktır. Bu tip algoritmalar kullanıldığında, uygulama (veya protokol) öyle tasarlanmalıdır ki saldırgan istediği düz metni şifrelenmiş olarak elde etmemelidir.

4.4.2.4. Ortadaki adam saldırısı:

Bu saldırı, kriptografik iletişim ve anahtar değişimi protokolleri ile ilgilidir. Fikir şudur; iki kişi güvenli iletişim için anahtarlarını değiş-tokuş ederken (örneğin Diffie-Hellman kullanarak), bir düşman kendisini iletişim hattındaki iki kişi arasına yerleştirir. Sonra bu düşman her iki kişi ile ayrı bir anahtar değiş-tokuşu gerçekleştirir. Her iki kişi farklı bir anahtar kullanarak işlerini tamamlayacaklardır ki bu anahtarlar düşman tarafından bilinmektedirler. Bu noktadan sonra saldırgan uygun anahtar ile herhangi bir iletişimi deşifre edebilecek ve bunları diğer kişiye iletmek için diğer anahtar ile şifreleyecektir. Her iki tarafta güvenli bir şekilde konuştuklarını sanacaklardır, ancak gerçekte saldırgan konuşulan her şeyi duymaktadır.

Ortadaki-adam-saldırısını engellemenin bir yolu dijital imzaları kullanabilen bir açık anahtar kripto sistemi kullanmaktır. Kurulum için her iki tarafta karşı tarafın açık anahtarını bilmelidir. Paylaşılan gizlilik oluşturulduktan sonra, taraflar kendi dijital imzalarını karşı tarafa göndermelidir. Ortadaki-Adam bu imzaları taklit etmeye çalışacak, fakat imzaların sahtesini yapamayacağı için başarısız olacaktır.

5. KAOS

"Bizim güzellik anlayışımız, düzen ve düzensizliğin ahenkli bir biçimde tertiplenmesinden" esinlenmektedir ki, bu doğal nesnelerde - bulutlarda, ağaçlarda, sıra dağlarda ya da kar tanelerinde - ortaya çıktığı şekliyle görülmektedir. Bütün bu nesnelerin şekilleri fiziksel biçim kalıplarına dökülmüş dinamik süreçlerdir; düzen ve düzensizliğin özel bir takım kombinasyonları da sadece bunlara özgüdür (Gleick, 1997:136).

Kaos kuramcılarına göre "kaos";
* bir durumun değil bir sürecin bilimi,
* bir varoluşun değil bir oluşumun bilimi
* sistemlerin global doğasının bilimidir ve
* karmaşıklığın evrensel davranış biçimini sorgular (Gleick 1997; Faigenbaum 1981, Prigogine & Stengers 1984; Cramer, 1998).

Bu düşünceyi ilk ortaya atanlardan biri olan Feigenbaum'a göre; bütün bilim alanlarının gelip gelip takıldığı bir sorun olan sistemlerin nonlineer davranış biçimi çözülmeliydi. Bu problemin temeline inebilmek için; bu teorilerin bilinmesi gerekiyordu. Bu konuyla ilgili diğer fizikçiler de Einstein'ın izafiyet (görecelik) kuramının bilinmesinin gerektiğini düşünüyorlardı. Bu teorinin katkısıyla; incelenen nonlineer olgu veya gelişmelerin ve türbülansların farklı ölçek ve boyutlarda değerlendirilebilmesi yaklaşımı ve "fraktal geometri" geliştirilmiştir.

Gökyüzünde hareket eden bulutlar, içilen bir sigaradan havaya yükselen duman bir musluktan akan su damlaları, bir rüzgarın etkisiyle oradan buraya salınıveren yapraklar... İlk bakışta her biri diğerinden farklı gibi görülen bu doğa manzaralarının içerdiği, bir ana tema vardır. Bütün bunlar ve yaşamımız boyunca her an tanık olduğumuz bir çok olay bize düzensiz, kuraldışı, devamsız olarak niteleyebileceğimiz oluşumlar olarak gözükür. Gerçekten de düzensiz ve her an öngörülemez bir geleceğe dönüşüveren doğa olayları, başta fizikçiler olmak üzere diğer doğa bilimcileri için bir sırdır. Bu özelliğinden dolayı, belirlenen tipte doğa

durumlarına "bilinemezlik-karışıklık" anlamında "kaos" adı verilir. Kaos terimi, yalnızca adı geçen oluşumları tanımlamakla kalmaz, aynı zamanda bunları bilimsel çerçevede incelemenin, anlamanın ve kontrol etmenin de adı olur.

J.Gleick. kaos için şunları yazmıştır:

"Kaosun başladığı noktada klasik bilim durur. Fizik bilimi varolduğundan beri doğanın yasalar; araştırılıp sorgulanmış ancak fizikçiler atmosferde, çalkantılı denizlerde, yabani popülasyonların dalgalanmalarında, kalp ve beynin titreşimlerinde varolan düzensizlik konusuna gelip dayandıklarında dünyayı bu konuların cahili olmaktan kurtaramamışlardır. Doğanın kuraldışı olan devamsızlık ve düzensizlik gösteren yüzü, bilim için bir bilmece olarak kalmıştır."

Kaosu ilgi çekici kılan ve bir devrim olarak değerlendirilmesine yol açan şey, kararlı olarak (belli sabit kanunlara göre) evrimleşen bir sistemin (hiç beklenmedik şekilde) düzensiz ve rast gele davranabilmesidir. Mesela bir adada yaşayan belli bir canlı türünün sayısı, bu canlının üreme hızına, adadaki besin miktarına, bu canlıyla beslenen diğer türlerin adadaki etkinliği vs. gibi birtakım faktörlere belli oranda bağlıdır. Benzer şekilde bir benzin istasyonuna uğrayan arabaların veya âcil servise gelen hastaların geliş ve servis zamanlarının dağılışı bu çerçevede değerlendirilebilir. Kolaylık olması için bütün bu faktörlerin zamanla değişmediği kabul edilerek yapılan en basit modellerde bile çok ilginç sonuçlar ortaya çıkmaktadır. Üreme hızını ifade eden bir katsayı belli bir değerden küçükse, çoğalma ve ölümleri ifade eden denklemler arasında bir denge kurulmakta ve nüfus belli bir değerde sabit kalmaktadır. Fakat bu katsayı, kritik bir değerden büyükse, nüfusun yıllara göre değişimi hiçbir formülle ifade edilemeyecek kadar düzensizleşmekte ve nüfus her yıl rast gele değerler alabilmektedir. Yani nüfus değişimi tamamen belirli kanunlara göre cereyan eden (deterministlik) bir hadise olmasına rağmen, sonuçta belirsizlik ve düzensizlik doğmaktadır. Benzeri bir durum, bir borudan akan sürtünmeli bir sıvının akış şekillerinde de görülmektedir. Akışkanların uyduğu dinamik denklemler Newton kanunlarından çıkartılıp düzgün akışlara gayet güzel bir şekilde uygulanabilmesine rağmen, akış hızı belli bir değeri aştığında girdaplar ve türbülans hareketleri oluşmakta ve akış tamamen kaotik hale gelmektedir.

Kaos teorisi engin uygulama alanına sahip olan bir yaklaşımdır. Her türlü alanda uygulanabilme yeteneğinden dolayı, kaos teorisinin bilim dallarını birbirinden

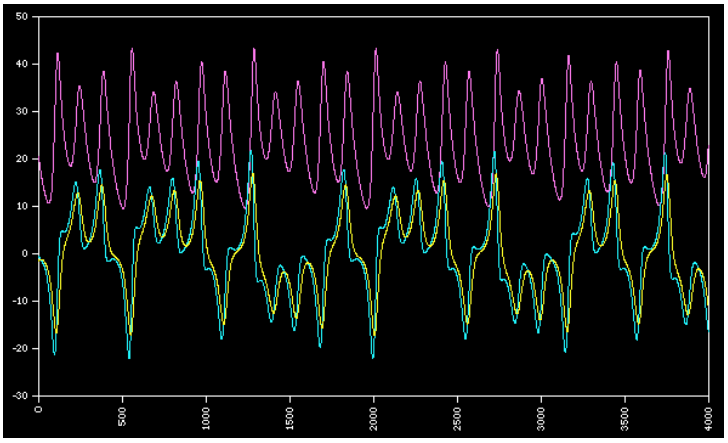
soyutlayan engelleri aştığı söylenebilir. Çok küçük görünen bir nedenin kendisinden çok daha büyük sonuçlara yol açabileceği mantığından hareket eden kaos kuramı, düzensizlik ve karmaşadan çok, bu düzensizlik içerisinde belli bir düzeni, düzenli düzensizliği anlamaya yöneliktir. Doğadaki şekillerin küp, koni, küre ve silindir gibi standart şekiller olmaktan çok, daha düzensiz görünümlü ve doğrusallıktan uzak şekiller olduğu noktasından hareket eden fraktal geometri kavramı da, kaos kuramı analizlerinde büyük faydalar sağlar.

Başlangıç durumuna hassas bağlılık

Kaosun 1960'lı yıllarda Lorenz ile başladığı söylenebilir. Lorenz kaosun en önemli kavramlarından birini, başlangıç durumlarına hassas bağımlılığını bulmuştur. Ayrıca Lorenz'in çekicisi(Lorenz Attractor) de uzun yıllar kaosu tanımlamıştı.

II. Dünya Savaşı sonunda. Massachusetis Teknoloji Enstitüsünde hava tahminleri üzerine bilgisayar destekli araştırmalarda bulunan E. Lorenz, 1960'ta icat ettiği minyatür meteoroloji modeliyle meslektaşlarını şaşırtmıştır. Lorenz ilkel bilgisayarını kullanarak havayı en basit şekilde ifade edilebilen bir hale indirgemiştir. Bilgisayarı, havadaki ısı-basınç ilişkilerini, rüzgarın yönünü, siklon gruplaşmalarını sayısal olarak, her gün 12 denklem yardımıyla kaydediyordu. Rüzgarlar ve hava sıcaklıkları, Lorenz' in yazıcısından satır satır dökülürken dünyadaki gerçekleşme biçimiyle aynı davranışı gösteriyordu. Dahası, bu listelenen değerlerden hareketle tahminlerde de "bulunabiliyordu. Biraz uğraşıdan sonra sayıları grafiğe de dökmeyi başardı.

1961'in kış aylarında, bu işi kestirme yoldan yapmak için, makineye önceki rapor değerlerini klavyeden girdi. Ancak rapordaki değerlerin sıfırdan sonraki üç rakamını yuvarlayarak. Şaşkınlıkla gördü ki, hava durumu, bir önceki dökümde yer alan şekilden umulmadık derecede sapmıştı. Bir süre sonra da dökümle hiç alakasız bir hava raporu grafiği elde etti.



Şekil 5-1 : Yukarıdaki şekilde Lorenz'in hava tahminlerini yapmak için kullandığı formüllerin sonucunda elde edilen iki ayrı grafik görülmektedir. Şekilde de görüldüğü gibi başlangıç değerlerinde yapılan küçük bir değişiklik sonuçta birbirinden farklı grafiklerin oluşmasına sebep olmaktadır.

Bu olay, Lorenz'e en küçük detayların bile ileride çok büyük sonuçlara neden olabileceğini hiç bir zaman uzun süreli bir hava tahmini yapılamayacağını göstermişti. Sistem, ilk çıkış noktasıyla çok hassas olarak ilişkiliydi. Bu olayı J.H.Poincare şöyle ifade ediyordu: "Başlangıç şartlarındaki küçücük bir hata son olguda muazzam bir hataya neden olacaktır. Bu durumda, olacağı öngörmek olanaklı değildir."

Kaotik sistemlerde, sistemin zaman içindeki gelişimini tam olarak belirleyebilmek için başlangıç değerlerini sonsuz hassasiyetle bilmek gerekmektedir. Herhangi bir dinamik sistem için de bu geçerlidir, fakat doğrusal (lineer) sistemlerde hata zamanla doğrusal artmasına rağmen kaotik sistemlerde üsse bağlı olarak artmaktadır. Yani doğrusal bir sistemde, mesela bir gezegenin Güneş etrafındaki hareketinde, başlangıçtaki gözlemde yaptığımız hata 1 birim ise, zamanla 2,3,4,... şeklinde artacak; fakat kaotik bir sistemde, mesela atmosferde 10, 100, 1000, 10 000,... şeklinde korkunç bir süratle artacaktır. Bunun sonucunda da uzun süreli hava tahminleri kesinlikle mümkün olmayacaktır.

Daha gelişmiş atmosfer modellerinin ve daha hızlı bilgisayarların kullanımıyla iyi tahminler yapılabileceğini düşünen bazı araştırmacılar, ABD'de *Control Data Cyber* 205 isimli bir süper bilgisayarı hava tahminleri için devreye

soktular. Lorenz'in ilkel bilgisayarı saniyede 60 işlem yapabilirken, yeni süper bilgisayarın hızı saniyede milyonlarca işlemle ölçülüyordu. Lorenz modelinde 12 denklemle yetinmesine rağmen Control Data Cyber'e 500.000 denklem yüklenmişti. Bu denklemlerde havadaki nemin yoğunlaşmasıyla meydana gelen sıcaklık değişimleri, sıradağların rüzgâr akımlarına etkisi de hesaba katılmıştı. Veriler tüm eyaletlerdeki uçaklardan, uydulardan ve gemilerden merkeze akıyordu. Bu şartlar altında dahi yapılan hava tahminleri iki veya üç günden sonra spekülatif olmaya başlamış, bir hafta sonra ise tamamen değerini kaybetmişti. Bunun sebebi kelebek etkisi idi. Kelebek etkisi, yarı şaka-yarı ciddi olarak Pekin'de kanat çırpan bir kelebeğin bir ay sonra New York'ta fırtınalar meydana getirebileceğini anlatmak için kullanılan bir deyimdir. Atmosfer kararsız, karmaşık ve kaotik bir sistem olduğu için herhangi bir hava tahmini belirli bir süre sonra değerini yitirmekte, çok küçük etkiler ve belirsizlikler katlanarak çığ gibi büyümektedir. Yanlış anlaşılmaması gereken nokta, kelebeğin oluşturduğu hava akımlarının enerjisiyle fırtınaların meydana gelmediğidir. Enerji zaten atmosferde mevcut olup yönlendirilmesi, yani değişik hava olaylarının oluşması, sistemin başlangıç şartlarına çok hassas bir şekilde bağlı olmaktadır. Bu konuda yapılan başka bir hesaplama ise kâinatın öbür ucundaki bir galaksideki bir elektronun çekim etkisinin yok edilmesi halinde dünyamızda üç-dört ay sonra büyük bir fırtınanın oluşabileceğini göstermiştir.

5.1 Lorenz Çekicisi

Massachusetts Teknoloji Enstitüsün'de meteoroloji uzmanı olan Edward Lorenz'in (Gleick, 1987), atmosferik konveksiyon (ısıyayım) olgusuna ilişkin yaptığı araştırma şöyledir: Güneş ışınlarının yeryüzünü ısıtması ve bu ısının havaya yansıması nedeniyle atmosferin alt katmalarındaki hava daha sıcak ve hafif duruma gelir. Isınan ve hafifleyen hava yukarı doğru yükselirken daha soğuk ve yoğun olan üst katmanlardaki hava aşağı doğru hareket eder. Bu iki yönlü harekete atmosferik konveksiyon denir. Lorenz atmosferde gerçekleşen bu hava hareketini kendi laboratuarında içinde gaz bulunan kapalı bir kutunun alt ve üst kısımlarına termometre yerleştirerek simüle etmiştir. Lorenz bu yöntemde iki eşitlik kullandığı halde aşağıdaki 3 eşitlik gerçek değerlere göre en iyi yaklaşımı vermiştir.

$dx / dt = \mu (y - x)$

$dy / dt = rx - y - xz$

$dz / dt = xy + bz$

Yukarıdaki üç diferansiyel eşitlik hava tahmininde kullanılan etkinlikleri temsil eden x, y ve z parametrelerinin değişimini tanımlamaktadır.

X : Konveksiyon (ısıyayım) sonucunda oluşan hava hareketinin hızı ile orantılıdır.

Y : Yükselen ve alçalan hava sıcaklıklarının farkı ile orantılıdır.

Z: Konveksiyona uğrayan havanın dikey olarak tepe ve alt kısımlarındaki sıcaklık farkı ile orantılıdır.

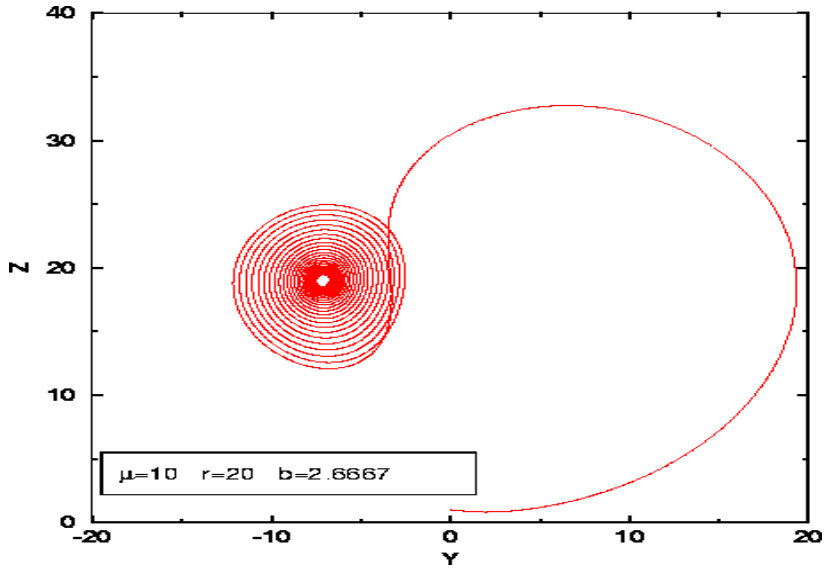
Denklemlerde kullanılan 3 ayrı sabit değer vardır. Bunlar:

μ: Havanın termal iletkenlik ve vizkozite(akışmazlık,koyuluk) değerlerinin birbirine oranını temsil eder. Bu sabit değer için genellikle 10 değeri kullanılır. Prandtl number olarak isimlendirilmektedir.

b : Hava konveksiyonun oluştuğu bölgenin genişliği ile yüksekliğinin birbirine oranını temsil eder. Lorenz'in bu sabit için sıkça kullandığı değer 2.6667 dır.

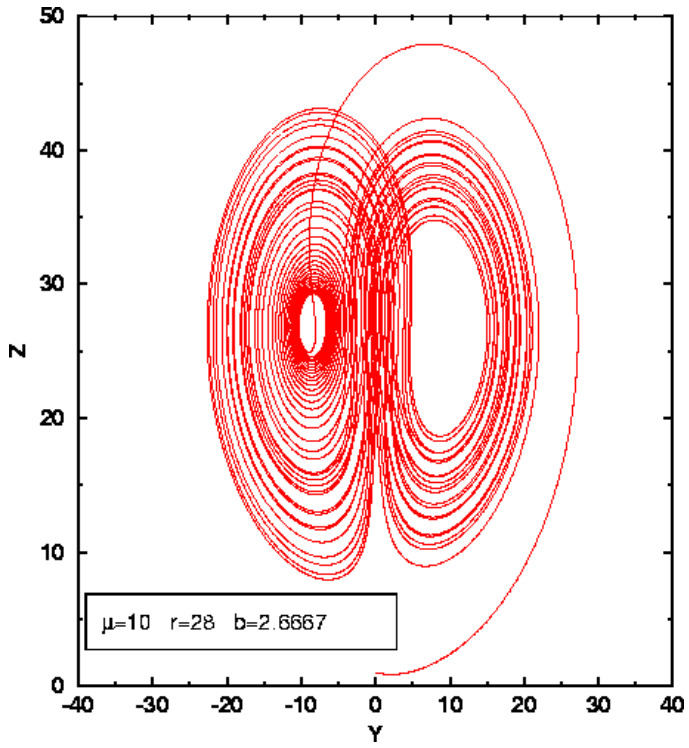
r : Sistemin taban ve tepe sıcaklıkları farkını temsil etmektedir. *Rayleigh number* olarak isimlendirilmektedir. Bu sabit değer için genellikle 20 değeri kullanılmaktadır.

Yukarıda açıklanan X, Y ve Z değişkenleri hava tahmini için gereken başlangıç değerleridir. Aşağıdaki örnek grafikte sabitler değerler için μ=10, b=2.6667 ve r=20 ve başlangıç değerleri için X=1, Y=0, Z=1 değerleri kullanılarak Y 'nin Z'ye göre değişimi gösterilmektedir.



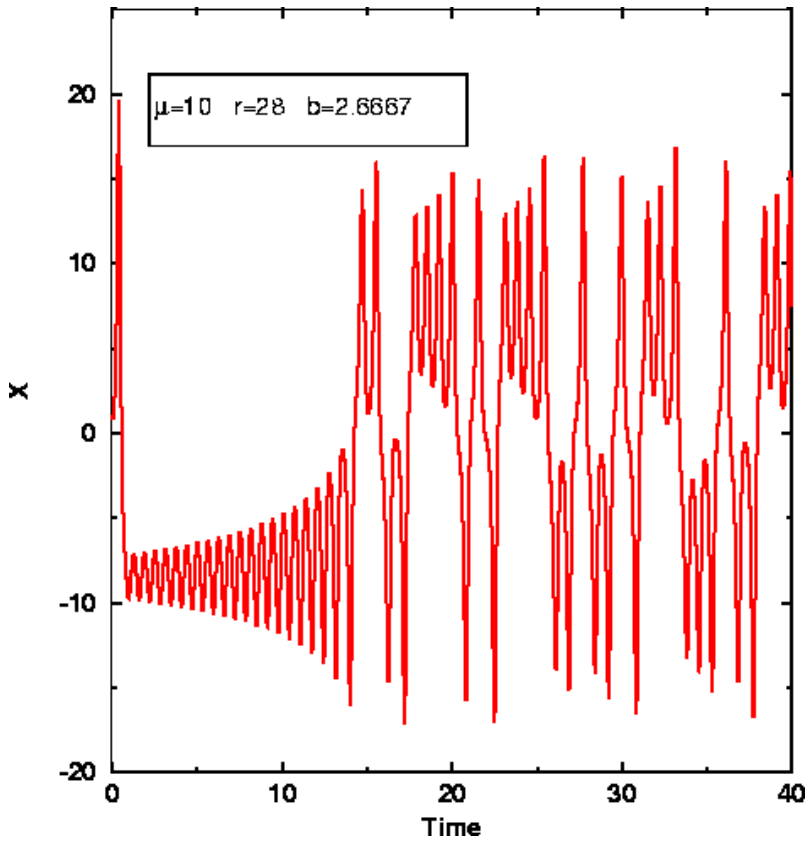
Şekil 5-2 : Artan zamana göre X ve Y, Y=-7.11, Z=18.99 noktasında sarmal bir şekil oluşturmaktadır.

Aşağıdaki şekil bu sefer sabit değerler için μ=10, b=2.6667 ve r=28 değerleri kullanılarak elde edilmiştir.



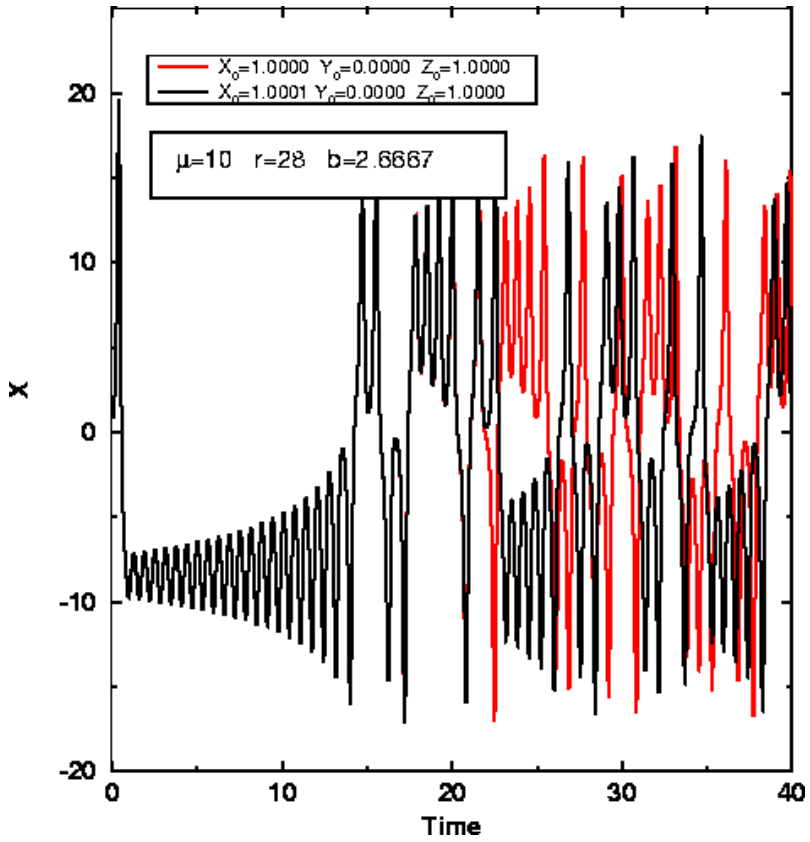
Şekil 5-3 : Yukarıdaki şekilde başlangıç değerleri şekil5-2 de kullanılan başlangıç değerleri ile aynı olmasına rağmen r sabit değerinin r=28 alınmasıyla tamamıyla farklı bir grafik elde edilmiştir.

Aşağıdaki şekilde ise şekil-3'e bağlı olarak X'in zaman göre değişimi görülmektedir.



Şekil 5-4 : X'in zaman göre değişim grafiği. Sabit değerler μ=10, b=2.6667, r=28 ve t=0 anında X=1 başlangıç değeri kullanılarak elde edilmiştir.

Şimdi ise aynı parametreler kullanarak fakat X=1 ve X=1.0001 değerleri için oluşacak grafiklere bakalım.

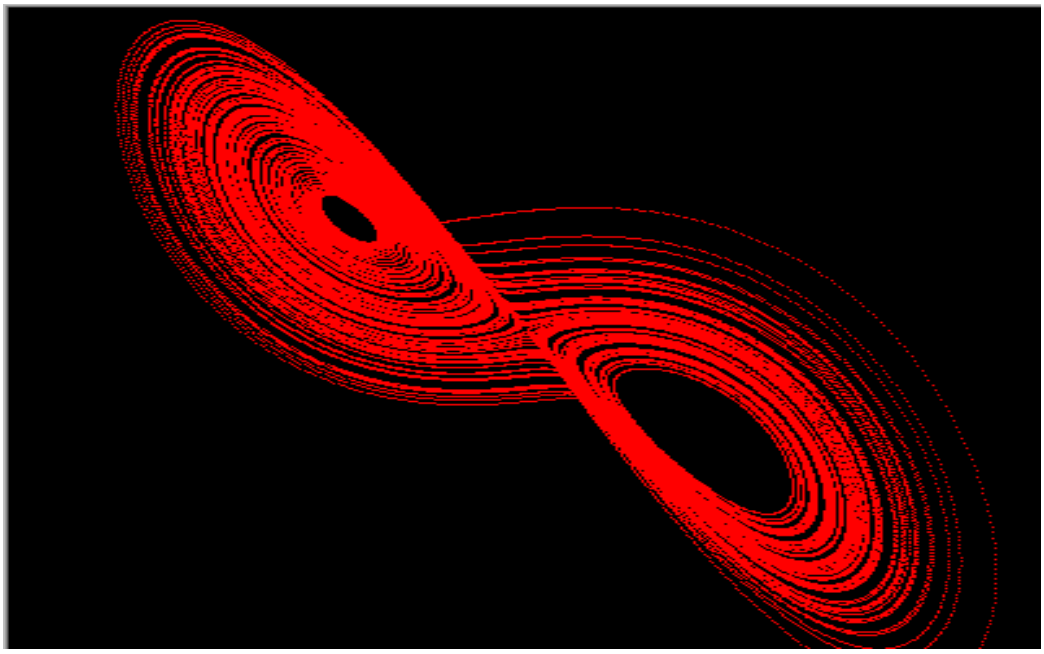


Şekil 5-5 : Şekilde μ=10, b=2.6667 ve r=28 sabit değerleri kullanılarak X=1 ve X=1.0001 başlangıç değerleri için iki ayrı grafik görülmektedir. Siyah ve kırmızı grafikler t=22 anından sonra farklılık göstermektedir.

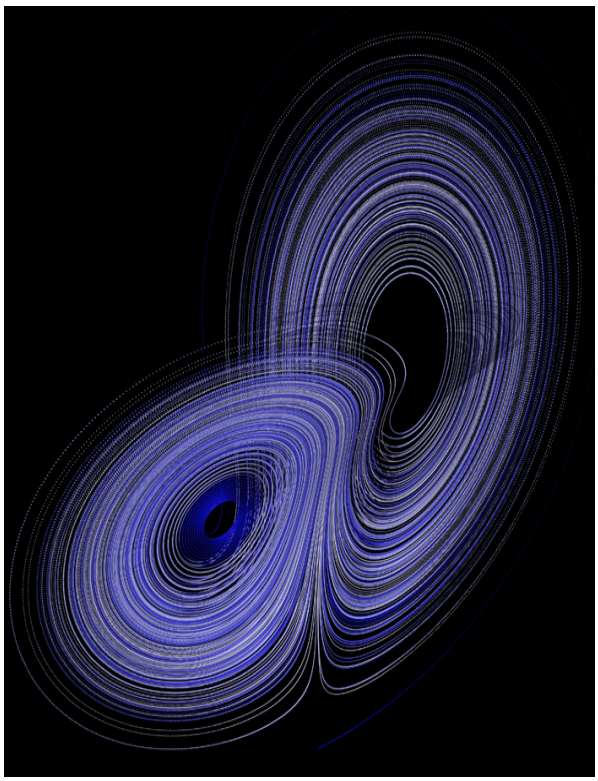
Yukarıdaki grafikte aynı sabit değerler kullanılmasına rağmen başlangıç değerlerindeki küçük bir farlılık sonucu (X=1 ve X=1.0001) t=22 anından sonra birbirinden tamamen farklı iki ayrı grafik elde edilmektedir. Şekil-3'te μ=10, b=2.6667 ve r=20 sabit değerleri kullanılmış fakat X=1 ve X=5 gibi birbirinden tamamen farklı başlangıç değerleri için hemen hemen aynı grafik elde edilmişti.

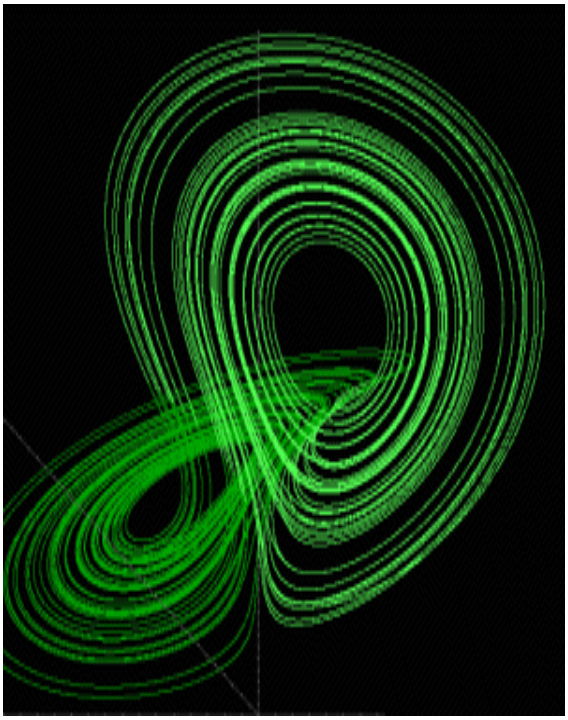
Bu sonuç bize kaotik bir problemin çözümünün başlangıç değişkenleri ve başlangıç sabitlerinin her ikisine birden sıkı ve hassas bir şekilde bağlı olduğunu göstermektedir. Başlangıç değerlerinde ve sabitlerinde yapılan küçük bir değişiklik sonuçta büyük bir değişiklik olmasına sebep olmaktadır.

Aşağıda Lorenz çekicisinin 3 boyutlu şekilleri görülmektedir.



Şekil 5.6 Başlangıç değerleri değiştirildiğinde farklı şekiller ortaya çıkmıştır.





Şekil 5-7 :Şekillerde de görüldüğü başlangıç değerlerinde ki farklılıktan dolayı değişik görünümlü şekiller elde edilmektedir.

5.2 Nonlineerlik ve Türbülans

Lineer denklemlerde, ilişkiler bir grafik üzerinde "doğru" ile ifade edilebilir. Nonlineer (doğrusal olmayan) denklemlerde ise, tanımlanan ilişkilerin birbirlerine orantılı olmaları gerekmektedir. *Nonlineer sistemler*, genellikle çözüme elverişli değildir ve bir araya getirilemez. Akışkan sistemlerde ve mekanik sistemlerde, nonlineer şartlar, genellikle devre dışı bırakılmak istenir. Örneğin, sürtünme olmasaydı bir topa vurmak için duyulan enerji miktarı basit bir lineer denklemle ifade edilebilirdi. Oysa, sürtünme işin içine girince bu ilişki karmaşık hale gelir, sürtünmeye sabit bir değer eklenemez çünkü, sürtünmenin değeri hıza, hız da sürtünmeye bağlıdır. Bu *karşılıklı etkileşim ile değişebilme özelliği, nonlineerliğin hesaplanmasını zorlaştırmakta; aynı zamanda da lineer sistemlerde bulunmayan bir davranış biçimi zenginliği yaratmaktadır*. Gleick'e göre; "nonlineer davranış biçimini analiz etmek bir labirentte gezinmeye benzer; ancak bu labirentte duvarlar attığınız her adıma göre şekil değiştirmektedir".

Türbülans konusu, kaos kuramcılarına göre; akışkanlar dinamiğinin en eski problemlerinden biridir. Ancak teorik fizik, türbülans konusuna adeta uzak durmayı tercih etmiştir, akışkanların düzenli davranış biçimlerine sahip oldukları yerlerle daha çok ilgilenmiştir. Türbülansta, düzgün bir akış parçalanıp sarmallara, burgaçlara veya girdaplara dönüşür. Düzensiz şekiller, akışkan cisimlerle katı cisimler arasındaki sınırı bozar. Enerji büyük ölçekli hareketlerden küçük ölçekli hareketlere doğru hızla akar. Türbülans, her ölçek düzeyinde ortaya çıkan düzensizliktir; büyük girdaplar içindeki küçük istikrarsızlıktır. İleri düzeyde sönüm özelliği vardır, yani enerjiyi akıtır ve sürtünme yaratır. Türbülans gelişigüzelleşmiş hareket demektir.

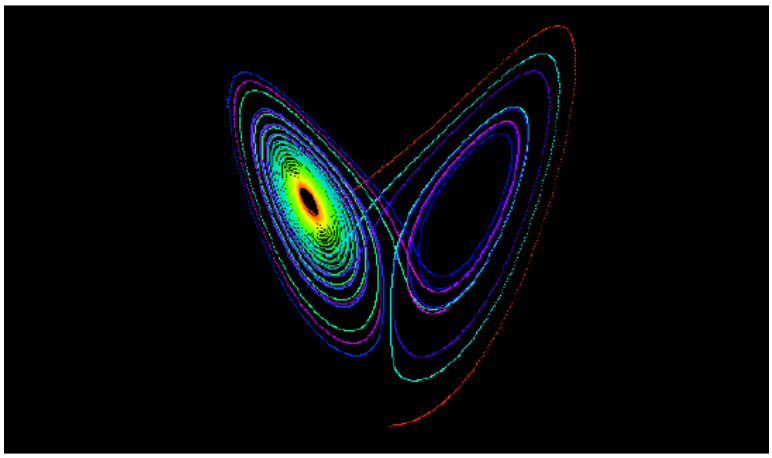
Türbülasyon halindeki bir girdaba daha yakından bakıldığında her ölçek düzeyinde, görüş alanınız içine yeni sükunet bölgeleri girdiği görülür. Bu nedenle, homojenlik varsayımı yerini kesintili varsayımına bırakır. Kesintili resim belirli bir şekilde ideal hale getirildiğinde, ileri düzeyde fraktal bir görünüm alarak, büyükten

küçüğe doğru hızla geçen ölçeklerde pürüzlü ve düzgün bölgeler birbiriyle karışık hale gelir. Aslında bu resim de gerçeği tam olarak yansıtmaz.

Bu konuda Rus bilim adamı Lev D. Landau'nun geleneksel paradigma olarak değerlendirilen modeli, birbiriyle mücadele eden ritimlerin üst üste yığılmasından oluşmuştur. Bu modele göre; bir sistemin içine daha fazla enerji dahil olduğunda, yeni frekanslar teker teker devreye girer ve bu frekansların her biri bir öncekiyle uyumsuzdur. (Gleick, 1997: 141-145).

Türbülans konusuna ilişkin herhangi bir kuramın "*başlangıç durumuna hassas bağlılığı*" da içermesi gerektiği mesajını veren Edward Lorenz'in "garip çekerler modeli" olmuştur. Aslında bir meteorolog olan Lorenz hem aperiodikliği hem de başlangıç durumuna hassas bağımlılığı meteoroloji olayında canlandırmıştır. Dünyanın herhangi bir yerinde, bir kelebeğin kanat çırpmalarının belli bir süre sonra atmosferin durumunu tümüyle değiştirebileceği görüşü "***Kelebek Etkisi***" olarak adlandırılan yeni bir kavramın doğmasına neden olmuştur.

Kelebek etkisi, sadece başlangıç durumuna hassas bağımlılığı değil, aslında bütünlüğün ve bu bütünlük içindeki oluşumların, ilişkilerin birbirine bağlılığının da kavramsallaştırması olmuştur. Günümüzdeki küresel ilişkilerde çok rahat izleyebileceğimiz olay ve ilişkilerin de açıklanmasında kullanılabilecek bir kavram haline gelmiştir.

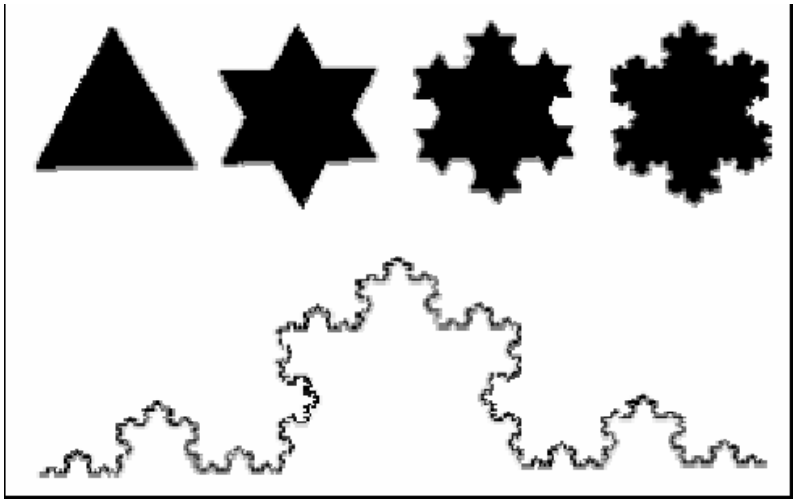


Şekil 5.8 Kelebek etkisi, Lorenz çekicisi

5.2.1 Kesir Boyut ve Fraktal Geometri

Kaos kuramcıları, "doğanın kendi doğasının ne kadar nonlineer olduğunu""düzensizliğin varolduğunu ve bu düzensizliğin ne olduğunun incelenmesi gerektiğini" vurgulamaktadırlar. Mandelbrot bu düzensizliği farklı ölçeklerde incelemek gerektiğini ileri sürmüştür.

O boyut konusuna da eğilmiş, "göreceli"ğe atıfta bulunarak bir nokta, doğru veya nesnenin uzaydaki yerini tanımlarken kullandığımız 0,1,2,3,... boyutlarının mikroskobik ölçeğe inildiğinde yetersiz kaldığını öne sürerek *"kesir boyut"* yaklaşımını getirmiştir. Kesir boyut başka türlü açıkça izahı yapılamayan "bir nesnedeki pütürlülük veya kırıklık yada düzensizlik derecesi gibi" nitelikleri ölçmenin bir yolu haline gelmiştir. Kıvrımlar halindeki bir sahil şeridinin ölçülmesi).



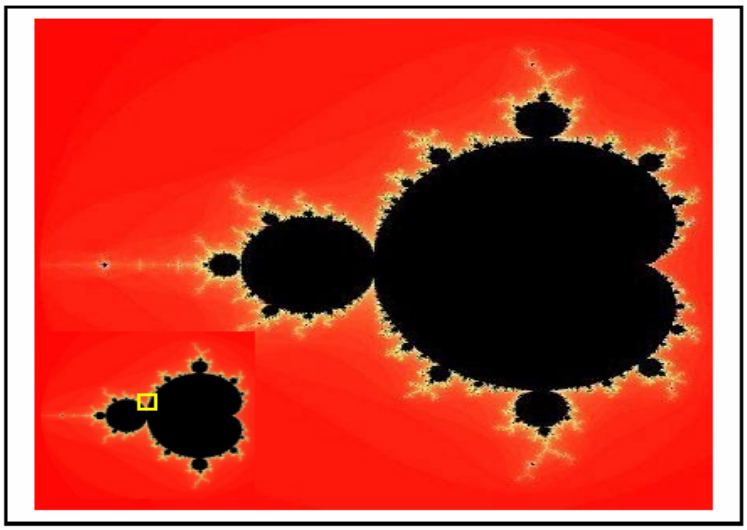
Şekil 5.9 Koch'un kar taneleri (Gleick, 1997:114)

Bir matematikçi olan Mandelbrot kesir boyut kavramından yola çıkarak geliştirdiği *"fraktal geometri"* sayesinde *"doğada incelediği düzensiz şekillerin*

ölçekler değiştiğinde düzensizlik derecesinin sabit kaldığını" öne sürmüştür. Buna göre; *"dünyamız düzenli bir düzensizliği sergilemektedir"*. (Aklın gözüne göre, bir fraktal sonsuzu görebilmenin bir yoludur). Mandelbrot'un, gerek düzensiz dokuları doğal süreçler içinde incelemesinin, gerekse sonsuz karmaşık şekilleri araştırmasının sonucunda aynı prensip bulunuyordu: "***kendi kendine benzerlik****"* niteliği. Zaten, "***fraktal***" "***kendine benzeyen***" anlamına gelmektedir. Kendi kendine benzerlik, her düzeydeki "*simetri*" demektir. İçeriğinde, başlanan yere geri dönülmesi, desen içinde desen bulunması vardır. Mandelbrot'un fiyat grafikleriyle nehir grafiklerinde kendi kendine benzerlik ortaya konmuştur; çünkü yalnızca ayrıntıları gittikçe daha ince ölçeklerde ortaya çıkarmakla kalmamış, ayrıntıları belirli bazı sabit ölçülerde de meydana çıkarmıştır. Koch eğrisi gibi şekiller kendi kendine benzerlik sergiler, çünkü büyük çapta büyütüldüğü zaman bile tamamen aynı görünümü muhafaza eder (Gleick, 1997: 119).

Fraktal kelimesi düzensiz ve parçalı, kırıklı ve kesikli şekilleri (kar tanelerinin billurumsu eğrilerinden galaksilerin kesintili tozlarına kadar) betimlemek, hesaplamak ve düşünmek için kullanılacak bir araç olmuştur.

Robert May ve James Yorke gibi araştırmacıların keşfettiği şekiller, düzenli davranışla -kaotik davranış arasındaki karmaşık sınırlar çerçevesinde, ancak büyük ve küçük ölçekler arasındaki ilişki itibariyle açıklanması mümkün olan şüphe edilemez düzenlilikler göstermiştir. Nonlineer dinamiğin çözümünde kilit rol oynayan yapıların da fraktal olduğu ortaya çıkmıştır. Çok farklı disiplinlerden araştırmacılar, Mandelbrot'un yeni geometrisinin "*tabiatın kendi geometrisi*" olduğuna inanarak çalışmalarında kullanmaya başlamışlardır.



Şekil 5.10 Mandelbrot kümesi: giderek küçülen ölçekler içinde yapılan bir yolculuk kümenin karmaşıklığının arttığını gösterir; bu karmaşıklıkta beliren parçalar kümenin tümüne benzemektedir (Gleick,1987:128)

Gleick'e göre; bilim adamları nesneleri parçalara ayırırlar ve her birine tek tek bakarlar. Atomaltı parçacıkların arasındaki etkileşimi incelemek isterlerse ikisini veya üçünü bir araya getirirler. Bu işlemde epeyi komplikasyon vardır. Halbuki, *kendi kendine benzeme gücü karmaşıklığın çok daha yüksek düzeylerinde başlar.* Bu, bütüne bakabilme sorunudur. (Gleick, 1997: 136). Fraktal geometri de bu bütüncül bakış sayesinde ortaya çıkabilmiştir.

Fraktal geometri mantığı ile yeni örgütlenme ilkeleri de belirlenmeye başlamıştır. *"Kendine benzerlik" karmaşık sistemlerde "kendi - kendine organizasyon" sürecinin önemli bir düzen prensibi olarak ortaya çıkmıştır.*

5.2.2 Dalgalanmalı Düzen

Bürüksel Okulu'ndan olan Prigogine ve Stengers geleneksel bilimi, "çoğunlukla kapalı sistemlerle ve küçük girdilerin daima aynı tarzda küçük sonuçları verdiği doğrusal ilişkilerle ilgili" görmüşlerdir. Prigogine'ci paradigma bakışları "düzensizlik, kararsızlık, çeşitlilik, dengesizlik, doğrusal olmayan ilişkiler (ki bu

ilişkiler içinde küçük girdiler çok büyük neticeleri başlatabilir) ve geçicilik - zamanın akışına karşı yüksek bir duyarlılık" konularına çevirmiştir (Prigogine ve Stengers, 1998: 13).

Prigogine'in termodinamik kanunlarına dayalı çalışmaları üzerine dayanan Brüksel Okulu'nun düşünceleri "*kapsamlı bir değişme teorisi*" oluşturmuştur. Termodinamiğin birinci kanunu "dünyanın enerjisi sabittir"; ikincisi ise, "dünyanın entropisi bir maksimuma doğru gider" şeklindedir. İzole bir sistemin entropisinin bir maksimuma doğru arttığı fikri, termodinamiği doğurmuş olan teknolojik problemden çok öteye geçer. Artan entropi kayıplarla eşanlı değildir ve artık sistemin içindeki tabii süreçleri belirtir. Bunlar sistemi, sonunda*, "maksimum entropi"* durumuna karşılık gelen termodinamik *"denge"*ye götüren süreçlerdir.

6. UYGULAMA

Bu uygulama hem online hem de offline durumlarda verilerin güvenli bir şekilde saklanmasını ve güvenli bir şekilde nasıl iletilebileceğini gösterebilmek için yapılmıştır. Uygulamada temel amaç hızlı ve güvenli bir şekilde verileri şifreleyebilen ve güvenli bir şifreleme algoritmasının yerine getirmesi gereken temel nitelikleri sağlayan bir şifreleme algoritması geliştirmekti.

Şifreleme algoritmasının aşağıdaki nitelikleri sağlaması gerekmektedir.

- Gizlilik – Bilgi istenmeyen kişiler tarafından anlaşılamaz.
- Bütünlük – Bilgi saklanması veya iletilmesi sırasında, farkına varılmadan değiştirilemez.
- Reddedilemezlik – Bilgiyi oluşturan ya da gönderen, daha sonra bilgiyi kendisinin oluşturduğunu veya gönderdiğini inkar edemez.
- Kimlik belirleme – Gönderen ve alıcı, birbirlerinin kimliklerini doğrulayabilirler.

Uygulamanın bu şartları sağlaması için kaotik modelle şifrelemeyi RSA şifreleme metodu ile birleştirerek güvenli bir şifreleme modeli oluşturmaya çalışıldı. Uygulamada Gizlilik ve bütünlük şartlarını kaotik modeller şifreleme tekniği kullanarak, reddedilemezlik ve kimlik belirleme şartlarını açık anahtarlı bir şifreleme metodu olan RSA şifreleme metodu ile sağlandı.

Kaotik modelle şifreleme de kullanılan başlangıç değerlerini dosyanın sonuna eklenerek RSA şifreleme algoritması ile şifrelendi. Bu; başlangıç değerlerinin açıktan gönderilmesi olayını engellemiş oldu hem de hem de reddedilemezlik ve kimlik belirleme şartlarını da sağlamış oldu. Bu uygulama elde edilen bu teknikle sayısal imza uygulamaları içinde uygulanabilir bir örnek durumuna geldi.

6.1 Kaotik Sistemler de Veri Şifreleme ve Şifre Çözme

Tezin başlarında kaosun birçok bilim dalına uygulanabilirliğinden bahsedilmiş ve bu yüzden kaos teorisi bilimler arası bir teori olarak nitelendirilmişti.

Bu projede kaos teorisini kriptoloji(şifreleme,gizleme) ile birlikte kullanacağız. Daha önceki bölümlerde veri şifreleme ve deşifre etme işlemleri için belirli büyüklükte anahtarların kullanıldığından bahsedilmişti.

Veri şifrelemesinde kullanılan şifreleme algoritmasının gücü anahtarın uzunluğuna ve anahtarın gerçekten tahmin edilmesinin zorluğuna bağlıdır. Bu nedenle kullanılan anahtarların sabit değerler olmasından çok rast gele ve belirli kurallara göre oluşturulması gerekmektedir.

Bu projede anahtarların üretilmesinde Lorenz'in kullandığı diferansiyel denklemler kullanılmıştır. Bu yöntem kaotik olarak üretilen anahtarların tahmin edilmesini ve kırılmasını güçleştiren bir yöntemdir. Lorenz'in diferansiyel denklemleri kullanılarak her bir üçlü iterasyon sonucunda başlangıç değerlerine bağlı olarak yeni bir sayı üretilmektedir. Tek bir şifreleme için üretilen tüm sayılar bir dizi içerisinde saklanmaktadır. Bu dizi anahtar (map) olarak tanımlanmaktadır ve bu dizinin büyüklüğü şifrelemede kullanılacak anahtarın toplam uzunluğunu belirtmektedir. Daha iyi anlaşılması açısından iterasyon sonucunda üretilen her bir değer şifrelemede kullanılacak olan anahtarın bir dişini oluşturmaktadır. Üretilen tüm değerler ise anahtarın tamamını oluşturmaktadır.

Şifrelemenin büyüklüğü şifrelemede kullanılan anahtarın boyutu ile tanımlanmaktadır. Yani bir dosya 64 byte bir map(anahtarın) ile şifrelenmiş ise bu şifreleme 64 byte'lık şifreleme olarak tanımlanmaktadır. Her bir iterasyon sonucunda 1 bytelık bir sayı oluşturulduğunda 64 byte'lık bir şifreleme için 64 kez üçlü iterasyon yapılacak demektir. Daha iyi anlaşılması açısından örnek vermek gerekirse:

Önceden belirlenen X_0, Y_0, ve Z_0 başlangıç değerleri ve μ=10, b=2.6667, r=28 sabit değerleri aşağıdaki formüllerde yerine yazıldığında edilen elde edilen değerler bir sonraki iterasyon için kullanılacak olan yeni değerlerdir. İterasyon için kullanılacak ilk değerler ise X_0, Y_0 ve Z_0 başlangıç değerleridir.

$$X_{n+1} = X_n + \mu\,(Y_n - X_n)$$

$Y_{n+1} = Y_n + rX_n - Y_n - X_n Y_n$

$Z_{n+1} = Z_n + X_n Y_n + bZ_n$

Oluşturulan yeni değerlerin bir sonraki iterasyonda kullanılabilmesi için sırasıyla

$X_n = X_{n+1}$

$Y_n = Y_{n+1}$

$Z_n = Z_{n+1}$

değer atamalarının yapılması gerekmektedir.

Her bir iterasyon sonunda üretilen Z değeri ise anahtarın bir byte'ını oluşturmaktadır. Ve bu değerler anahtar(map) dizisi içerisine sırayla yerleştirilmektedir. Anahtarın boyutuna N byte dersek yukarıdaki iterasyon işlemi N kez tekrarlanarak anahtarın tamamı(map) oluşturulmaktadır. Başlangıç değerleri rast gele olabileceği gibi belirli değerlerde olabilir. İlerleyen kısımlarında anlatılacak olan ve bu proje için başlangıç değerleri dışarıdan kullanıcı tanımlı olarak girilebildiği gibi rast gele (*random*) olarak da üretilmektedir yani başlangıç değerinin oluşturulması şifrelemeyi yapacak olan kullanıcının kendi seçimine kalmıştır. Anahtarın boyutu ise kullanıcının ne kadar güçlü ve uzunlukta bir şifreleme yapmak istediğine bağlı olarak dışarıdan kullanıcı tanımlı olarak alınmaktadır.

Şimdi ise anahtarın üretilmesi işleminden sonra şifrelenecek olan verinin yada dosyanın bu anahtar kullanılarak nasıl şifrelendiğini açıklayalım. Şifreleme için bir çok algoritma geliştirilmiştir ve bu algoritmalarda da değişik teknikler kullanılmaktadır. Bu projede ise dosya şifrelemesi için Exclusive OR (XOR) işlemi kullanılmaktadır. Anahtar boyutuna bağlı olarak dosyanın tamamı anahtar değeri ile XOR işlemine tabi tutulmaktadır. Örnek vermek gerekirse anahtarın uzunluğu 32 byte olsun. Bu durumda dosya ilk byte'ından itibaren sonuna kadar 32 byte'lık bloklar halinde anahtar değeri ile XOR işlemine tabi tutularak şifrelenecektir. Tabi aynı anda sadece 1 byte'lık veri XOR işlemine tabi tutulabileceği için her bir 32 byte'lık blok şifreleme 32 kez yapılan XOR döngüsü neticesinde gerçekleşmektedir.

Şifrelenen verinin başkaları tarafından açılmasını önlemek için yapılması gereken en önemli tedbir anahtar üretimi için kullanılan başlangıç değerlerinin saklanmasıdır. Eğer bu başlangıç değerleri veriye ulaşmasını istemediğimiz kişilerin eline geçerse ve tabi bu kişi dosyanın nasıl bir teknik kullanılarak şifrelendiği hakkında bilgi sahibi ise veriyi deşifre etmek için pek fazla çaba göstermeyecektir. Başlangıç değerleri çok hassas bir yuvarlama hatası ile iterasyona sokulduğunda anahtar değeri(map dizisi) kolaylıkla elde edilebilecektir.

Kaotik veri şifreleme yönteminde anahtar değeri başlangıç değerlerine çok hassas bir şekilde bağlılık göstermektedir. Şifrelenmiş bir dosyanın kırılması anahtar değerinin elde edilmesini gerektirir bu ise anahtar üretiminde kullanılan başlangıç değerlerinin çok hassas bir şekilde tahmin edilmesini gerektirmektedir. Dosyanın deşifre edilmesi için kullanılacak anahtar değerlerinde yapılan küçük bir yuvarlama hatası bile tamamen farklı bir anahtarın üretilmesine sebep olacaktır ve sonuç itibariyle dosya deşifre edilemeyecektir. Bu sonuç ise kaotik veri şifreleme yönteminin gücünü ispatlamaktadır.

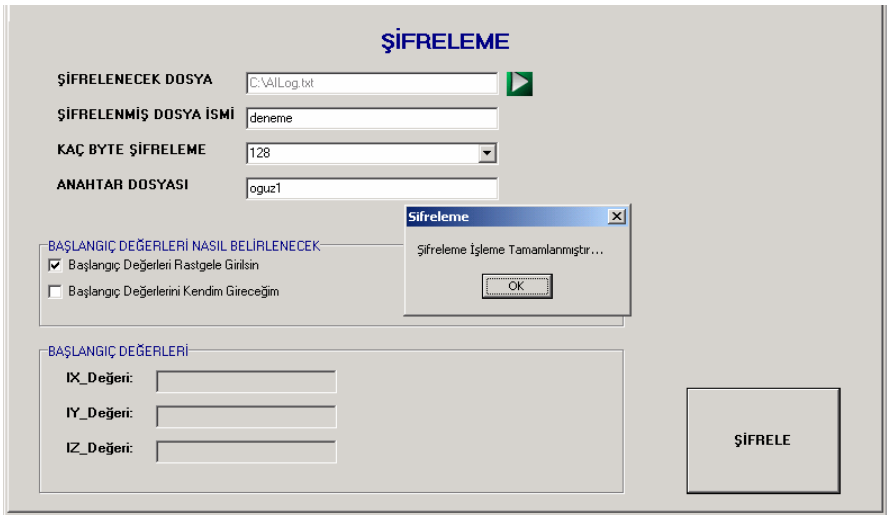
Veri şifrelendikten sonra tekrar eski haline getirmek için başlangıç değerlerine bağlı olarak yeniden aynı iterasyonlar gerçekleştirilir ve tekrar veri istenilen bloklar halinde XOR işlemine tabi tutularak orijinal veri elde edilir.

Bilindiği gibi kaotik şifreleme başlangıç değerlerine çok hassas olduğu için başlangıç değerlerinin güvenli bir şekilde korunması gerekmektedir. Bizde yapmış olduğumuz programda RSA ile başlangıç değerlerini şifreleyerek ayrı bir şekilde anahtar dosyasını oluşturduk. Bu şekilde yabancı kişilerin anahtarı elde etmesini önlemiş olduk. Anahtar dosyasına ulaşsa bile özel ve genel anahtarları bilmeden başlangıç değerlerini elde etmesi imkansız hal alacaktır.

Yapılan uygulama 6 bölümden oluşmaktadır. Sırasıyla

- Şifreleme
- Şifre Çözme
- Grafik
- Grafik Karşılaştırma

6.2 Şifreleme



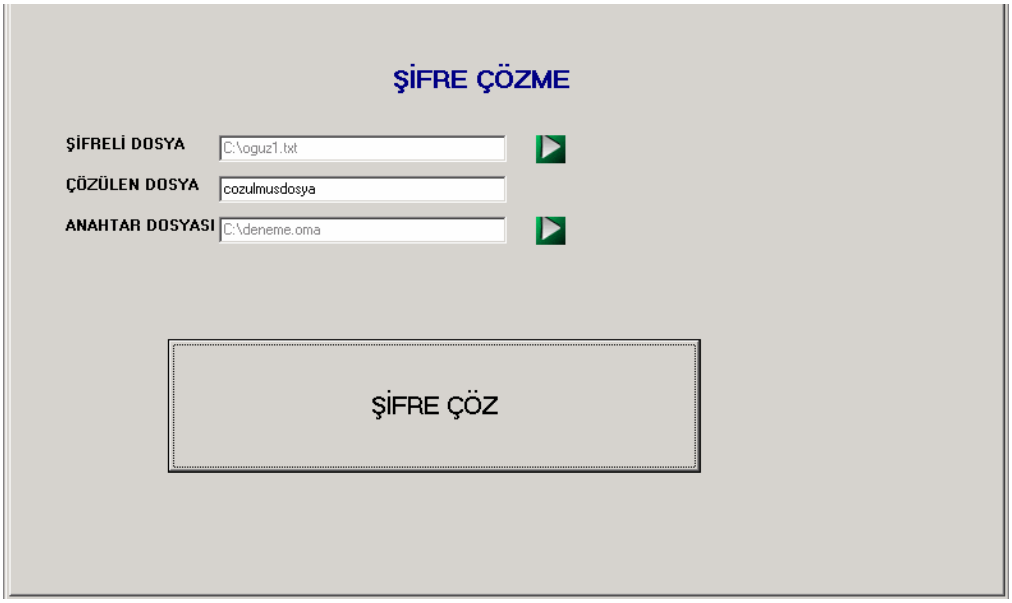
Şekil 6.1 Şifreleme

Bu kısımda text dosyaların kaotik şifreleme kullanarak şifrelenmesi gerçekleştirilmektedir. Şifrelenecek dosya ok tuşuna basılarak seçilir. Daha sonra şifrelenecek dosyanın ismi ne olacaksa Şifrelenmiş Dosya giriş kısmına yazılacaktır. Kaotik şifreleme aynı zamanda bir blok şifreleme türü olduğundan dolayı kaç byte şifreleneceği kullanıcıya bırakılmıştır. Burada 2 ve 128 arasında değişik şifreleme seçenekleri kullanıcıya bırakılmıştır. Kaotik şifreleme de anahtar değerlerini Lorenz diferansiyel denklemlerine göre gerçekleştirmiştik.Bu denkleme ilk x,y,z değerlerini vermek uygulamamızda iki yolla gerçekleştirilmektedir.

- Başlangıç Değerlerinin Rast gele Girilmesi
- Başlangıç Değerlerinin Kullanıcı Tarafından Girilmesi

Başlangıç değerleri seçildikten sonra yukarda ki seçeneklerden herhangi biri seçilerek şifreleme gerçekleştirilir. Şifrele butonu tıklandığında şifreleme gerçekleşecektir. Örneğin şifrelenmiş dosya ismine şifreli adını vermiş olsaydık, C:\oguz1.oma adı altında şifreli dosya oluşturulacaktır.

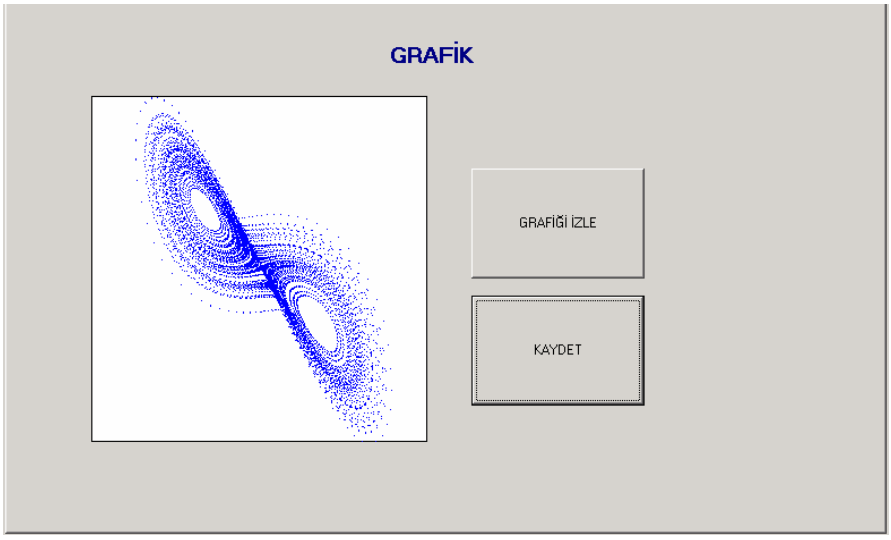
6.3 Şifre Çözme



Şekil 6.2. Şifre Çözme

Bu bölümde ise şifreli dosyayı eski normal haline getiriliyor. Bir önceki şifrelenmiş dosya seçilir. Çözülen dosya ismini kullanıcı girer. Burada dikkat edilmesi gereken anahtar dosyasıdır. Şifreleme Bölümünde başlangıç değerleri anahtar adlı dosyada RSA ile şifrelenerek saklanmıştı. Bu C:\ Anahtar.txt dosyası adı altında saklanmıştı. Anahtar dosyası 4 değişkeni içermektedir. Bunlar başlangıç değerleri olan X,Y ve Z değerleri ve kaç byte şifreleme yapılacağıdır. Bu değerler genel anahtar ile şifrelendikten sonra anahtar dosyasına kaydedilmektedir. Şifre çözme işlemini gerçekleştirilirken bu anahtar dosyasında yer alan değerler özel anahtar ile eski haline getirilmektedir.

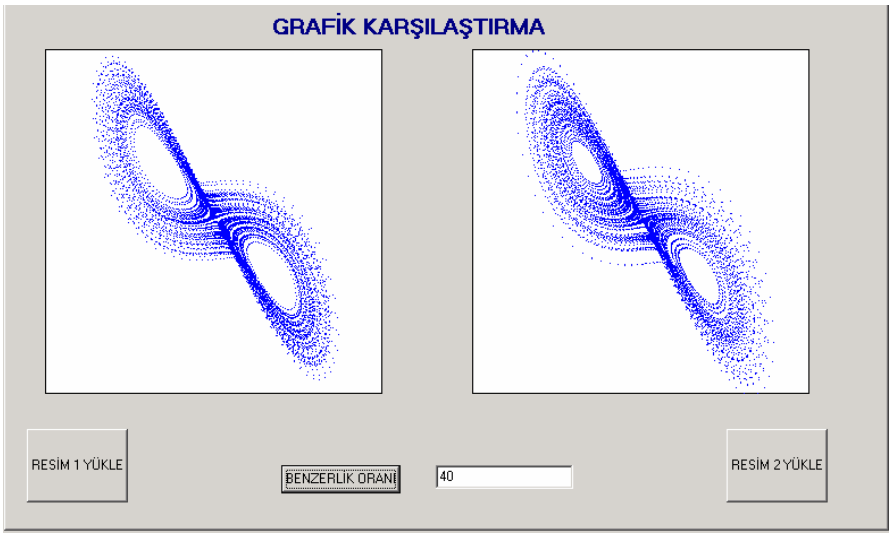
Şifre Çöz butonu tıklandığında C:\ dizininin altına çözülmüş dosyayı seçmiş olduğumuz isim altında atacaktır.



Şekil 6.3. Grafik oluşturma

Bu bölümde ise şifrelemeye başlamadan önce başlangıç değerlerini girilmişti. Bu başlangıç değerlerine göre Lorenz Eğrisinin fraktalını grafiği izle butonu kullanılarak izlenebilir. Bu grafiği ayrıca kaydet butonu kullanarak istediğimiz yere resim dosyası şeklinde kaydetmemiz mümkündür. Burada kaydetme seçeneğinin konulmasının sebebi bir sonraki bölüm olan grafik karşılaştırma bölümünde bu grafiklerin karşılaştırılmasına olanak sağlamaktır.

6.4. Grafik Karşılaştırma

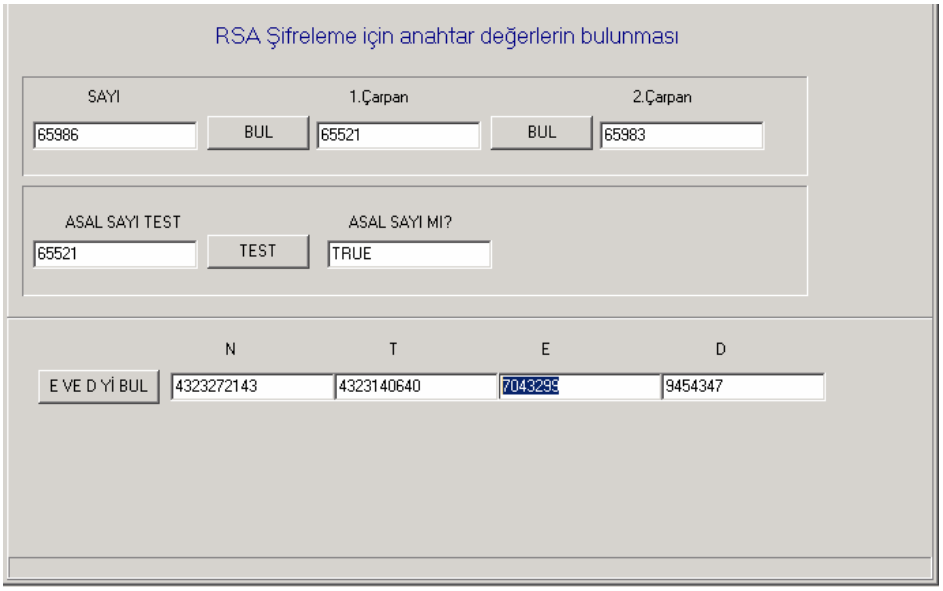


Şekil 6.4 Grafik karşılaştırma

Bu bölümde ise değişik başlangıç durumlarına göre meydana gelen Lorenz çekicisinin grafiklerinin birbirlerine ne kadar benzediğini öğreniriz. Bu grafiklerin birbirine ne kadar benzediği piksel piksel incelenmiştir. Bu grafiklerin benzerlik oranlarından başlangıç değerlerinin değişmesi ile meydana gelen anahtar değerleri arasında nasıl bir benzerlik olduğunu keşfetmiş olunur.

Anahtar oluşturma bölümü:

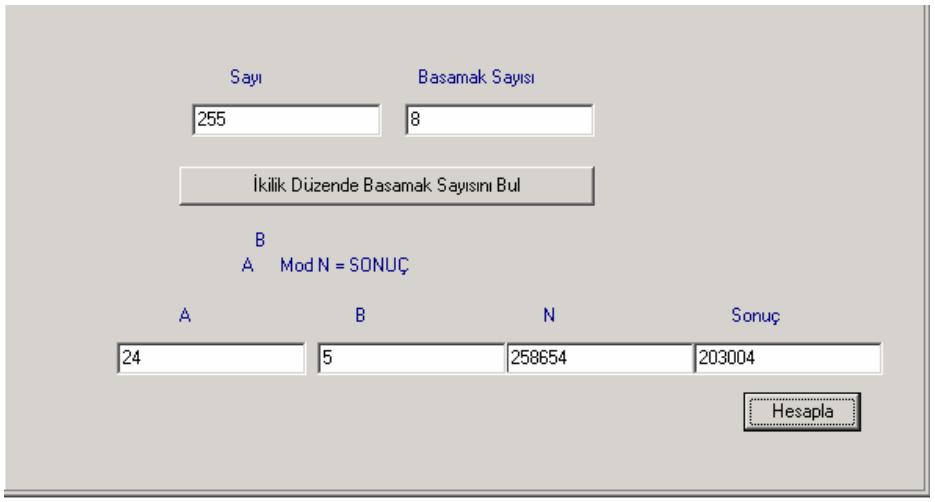
Bu bölümde Kaotik modelle verilerin şifrelenmesi gerekli olan başlangıç değerlerini RSA şifreleme algoritması ile şifrelemek için gerekli olan asal sayılar ve anahtarlar bulunuyor.



Şekil 6.5 Anahtar Oluşturma

Mod Bulma bölümü:

Bu bölümde bulunan e ve Qn değerlerine göre mod almanın örnek bir ekranı gösterilmiştir. Mod alma işlemi daha hızlı olması için üssün ikili düzene çevrilip sayının üssün değerine göre sola kaydırılıp modu alınarak hesaplanıyor.



Şekil 6.6. Mod bulma

7. SONUÇLAR ve ÖNERİLER

Başlangıç şartlarının bilinmesi durumunda nasıl davranacağı tamamıyla bilinebilen aksi halde ise rasgele davranışlar sergileyen kaotik sistemlerin başlangıç şartlarına aşırı duyarlılık göstermesi ve bu sayede birbirinden farklı çok sayıda dizinin elde edile bilmesi ve kriptografide kısa bir anahtar yardımıyla "key"uzun rast gele diziler üretebilen algoritmalara duyulan ihtiyaç, araştırmacıları kriptografide kullanılan anahtar dizilerinin ("key stream"), bu tür sistemlerin ürettiği dizilerle yer değiştirebileceği konusunda çalışmaya yöneltti.

Gerçek mesajın,rasgele sayılardan oluşan ve mesaj ile eş uzunluklu bir anahtar dizisiyle XOR işlemi kullanılarak şifrelenmesi anlamına gelen ve kırılamaz olduğu kanıtlanabilir tek zamanlı sistemler, tek seferlik kullanılan uzun rasgele güvenli bir şekilde dağıtılmasından kaynaklanan zorluklardan dolayı yaygın bir kullanım alanı kazanamadı. Kaotik sistemler bu probleme pratik bir çözüm getirebilirler. Çünkü kaotik sistemlerin parametrelerine ve başlangıç koşullarına duyarlılığından dolayı, ayar parametrelerindeki ufak değişimlerle birbirinden dolayı birbirinden tamamen farklı çok sayıda dizi üretmek mümkündür. Buna ilave olarak kaotik sistemlerin çok küçük hatalara duyarlılığından dolayı, dönüşümün tekrarlama değerinin tam olarak bilinmemesi durumunda anahtar dizisinin tümünün doğru olarak yeniden hesaplanmasına imkan yoktur. Daha fazlası şimdiye kadar yapılan çalışmalar, kaotik sistemlerin kriptografide sanki rasgele dizi üreteci olarak kullanımını destekler yöndedir.

Lorenz kaotik sistemi ile birbirlerinden farklı olan ilk koşul değerleriyle oluşturulan dizilerin birbirlerinden tamamen farklı olması, rasgele sayı üretmek için noktadan sonraki elverişli rakamların çokluğu dizilerin meydana getirebilecekleri çevrim uzunluklarının oldukça büyük olması ve x,y ve z değerlerinin farklı şekillerdeki pek çok kombinasyonu için rasgelelik testlerini geçebilen dizilerin oluşturulabilmesi ;bu sistemlerin rasgele sayı üreteci sistem olarak kullanılmasını desteklemektedir.

Yapılan incelemeler neticesinde, gizliliğin son derece önemli olduğu ve gelişen teknoloji ile birlikte bilgi hırsızlığının hat safhaya ulaştığı günümüzde, kaotik

sistemlerin ilgi odağı haline gelmesi ve diğer alanlara aktarılması, kaçınılmaz bir ihtiyaç olarak görünmektedir.

8. KAYNAKLAR

1. Edward Lorenz J.Atmos.S.20(1963) 130

2. James Gleick,Kaos,1995,Tubitak Press

3. Apohan, 1993, Kriptografi (*Cryptography*), Yüksek Lisans, İTÜ

4. Erhan, 1993, RSA algoritmasını kullanan şifreleme/deşifreleme yazılımının tasarımı (*Data encyption/decryption methods and software design of RSA algorithm*), Yüksek Lisans, İTÜ

5. Stinson, 1995, Cryptography Theory and Practice, CRC Press

6. Schneier, 1996, Applied Cryptography, John Wiley & Sons

7. Yiğit 1997, Çok-Boyutlu kaotik sistemler ile şifreleme,Yüksek Lisans,İTÜ

8. Develi 1997,güvenli haberleşme için kaotik işaretlerin ve sistemlerin incelenmesi, Yüksek Lisans,Erciyes Üniversitesi

9. Gül, 1997, RSA tabanlı halka açık anahtarlı kripto sisteminin uygulanması (*Implementation of on RSA based cryptosystem*), Yüksek Lisans, ODTU

10. Naccache, 1999, Padding Attacks on RSA, *Information Security Technical Report*, Vol.4, No.4, 28-33

11. Golic, 1999, Stream cipher encryption of random access files, *Information Processing Letters*, 145-148

12. Wright, 1999, The Evolution of the Advanced Encryption Standard, *Network Security*, 11-14

13. Aslan, 1999, Sayısal imza sistemlerinin incelenmesi (*Digital signature schemes*), Yüksek Lisans, İTÜ.

14. Saka, 2000, Bilgisayar ağ güvenliği ve şifreleme (*Network security and cryptography*), Yüksek Lisans, MUĞLA

15. Mingkai Nan , Chak-nam Wong , Kim-fung Tsang , Xiangquan Shi, Secure digital communication based on linearly synchronized chaotic maps,Letters A,2000

16. Goce Jakimoski, Ljup¡co Kocarev , Analysis of some recently proposed chaos-based encryption algorithms,November 2001

17. Chang, Hwang, Chen, 2001, A new encryption algorithm for image cryptosystems, *The Journal of Systems and Software*, 83-91

18. Peyravian, Matyas, Roginsky, Zunic, 2001, Alternative Method for Unique RSA Primes Generation, *Computers&Security*, Vol.20, No.2, 173-179

19. Torrubia, Mora, Martı, 2001, Cryptography Regulations for E-commerce and Digital Rights Management, *Computers&Security*, Vol.20, No.8, 724-738

20. Qurban A. Memon, Synchronized choas for network security,July 2002

21. A. Palacios , H. Juarez, Cryptography with cycling chaos,Letters A ,2002

22. P. García , J. Jiménez, Communication through chaotic map systems,Let. A,2002

23. Halife Kodaz 2002, Veri İletiminde Güvenlik İçin Şifreleme, Yüksek Lisans,Selçuk Üniversitesi

24. Xiaogang Wu, Hanping Hu, Baoliang Zhang, Analyzing and improving a chaotic encryption method,Chaos solitons&fractals,February 2004

25. NSA - The National Security Agency
(http://www.nsa.gov/about_nsa/faqs_internet.html)

26. Keys to Secret Drawers: The Clipper Chip and Encryption
(http://www.stardot.com/~lukeseem/j202/essay.html)

27. http://www.olympos.org, 20.03.2004

28. http://web.ansi.org/,09.04.2004

29. http://www.nist.gov/,10.04.2004

30. http://e-kimlik.bilten.metu.edu.tr/,14.04.2004

31. http://security.nyg.ege.edu.tr/, 15.04.2004

32. http://www.rsa.com, 17.04.2004

EK A

ÇOK-BOYUTLU KAOTİK SİSTEMLER İLE

RASGELE SAYILAR ÜRETEN ALGORİTMALAR

A.1 Chua Sistemini Kullanarak Rasgele Sayılar Üreten Algoritma

```
#include <stdio.h>
#include<stdlib.h>
#include<math.h>
#include<conio.h>
FILE *fp1;
Main()
{
double vc10=.1;
double i10 =.1;
double vc20=.2;
double vc1;
double il;
double vc2;
double alfa =9.0;
double beta =14.28;
double delta=.85;
int iz;
int a[16];
double m0=2.14285714285714285714;
double m1=1.71428571428571428571;
double m2 = -1;
double h=.1;
double k1;
char str1[35];
char str2[35];
char str3[35];
char st1[4];
char st2[4];
```

```
char st3[4];
double r1;
double r2;
double r3;
double r;
double i;
int as;
int ps=0;
int pt =0;
int pk=0;
clrscr();
fp1=fopen("oguz.dat","wb");
 if (fp1==NULL){
            printf("\oguz.dat dosyası açılamadı");
            exit(1);
            }
vc10=vc10-0.00001;
i10=i10+0.00001;
vc20=vc20-0.00001;
printf("\n%1f%1f%1f",vc10,vc20,i10);
for(i=0.0;i<131072.0;i++){
          if((vc10<=1.0)&&(vc10>=-1.0)){
               k1 = -alfa*(1+delta)*vc10+alfa*vc20+alfa*m0*vc10;
               vc1=vc10+h*k1;
            }
           if((vc10<-1.0)&&(vc10>=-20.0)){
               k1=-alfa*(1+delta)*vc10+alfa*vc20+alfa*(m1*(vc10+1)-m0);
               vc1=vc10+h*k1;
               }
           if((vc10<-20)&&(vc10<=-54.714285)){
               k1=-alfa*(1+delta)*vc10+aolfa*vc20+alfa*(m2*(vc10+20)-34.714285);
               vc1=vc10+h*k1;
```

```
}
vc2=vc20+h*(vc10-vc20+il0);
il=il0-h*beta*vc20;
sprintf(srtr1,"%.201f",vcl);
sprintf(srtr2,"%.201f",vc2);
sprintf(srtr3,"%.201f",il);
while(str1[ps]!='.')
      ps++;
if(str1[ps]=='.')
      ps++;
st1[0]=str1[ps+10];
st1[1]=str1[ps+11];
st1[2]=str1[ps+12];
st1[3]='\0';
while(str2[pt]!='.')
      pt++;
if(str2[pt]=='.')
      pt++;
st2[0]=str2[pt+10];
st2[1]=str2[pt+11];
st2[2]=str2[pt+12];
st2[3]='\0';
while(str3[pk]!='.')
      pk++;
if(str3[pk]=='.')
      pk++;
st3[0]=str3[pk+10];
st3[1]=str3[pk+11];
st3[2]=str3[pk+12];
st3[3]='\0';
```

```
            r1=atof(st1);
            r2=atof(st2);
            r3=atof(st3);
            r=r1+r2+r3;
            r=fmod(r,256.0);
            as=(int)(r);
            for(iz=15;iz>=0;iz--){
                if((as>>iz<<15))
                    a[iz]=1;
                else
                    a[iz]=0;
                }
            for(iz=7;iz>=0;iz--){
                fprintf(fp1,"%d",a[iz]);
                }
            vc10=vc1;
            il0=il;
            vc20=vc2;
            ps=0;
            pt=0;
            pk=0;
            }
fclose(fp1);
return 0;
}
```

A.2 Lorenz Sistemini Kullanarak Rasgele Sayılar Üreten Algoritma

```
#include <stdio.h>
#include<stdlib.h>
#include<math.h>
#include<conio.h>
FILE *fp1;
Main()
{
double sigma=10.0;
double r =28.0;
double b=2.66666666666;
double h=.01;
double x0=.1;
double y0=-.193;
double z0 =.701;
double x1;
double y1;
double z1;
double i;
char str1[35];
char str2[35];
char str3[35];
char st1[4];
char st2[4];
char st3[4];
double p1;
double p2;
double p3;
double p4;
int p;
int ps=0;
int pt =0;
```

```
int pk=0;
int iz;
int a[16];
clrscr();
x0=x0-0.01;
y0=y0+0.01;
z0=z0-0.01;
fp1=fopen("oguz.dat","wb");
 if (fp1==NULL){
            printf("\noguz.dat dosyası açılamadı");
            exit(1);
            }
for(i=0.0;i<131072.0;i++){
        x1=x0+h*(-sigma*x0+sigma*y0);
        y1=y0+h*(-x0*z0+r*x0-y0);
        z1=z0+h*(x0*y0-b*z0);
        sprintf(srtr1,"%.201f",x1);
        sprintf(srtr2,"%.201f",y1);
        sprintf(srtr3,"%.201f",z1);
        while(str1[ps]!='.')
            ps++;
        if(str1[ps]=='.')
            ps++;
        st1[0]=str1[ps+10];
        st1[1]=str1[ps+11];
        st1[2]=str1[ps+12];
        st1[3]='\0';
        while(str2[pt]!='.')
        pt++;
        if(str2[pt]=='.')
        pt++;
          st2[0]=str2[pt+10];
```

```
st2[1]=str2[pt+11];
st2[2]=str2[pt+12];
st2[3]='\0';
while(str3[pk]!='.')
      pk++;
if(str3[pk]=='.')
      pk++;
st3[0]=str3[pk+10];
st3[1]=str3[pk+11];
st3[2]=str3[pk+12];
st3[3]='\0';
p1=atof(st1);
p2=atof(st2);
p3=atof(st3);
p4=p1+p2+p3;
p4=fmod(p4,256.0);
p=(int)(p4);
for(iz=15;iz>=0;iz--){
    if((p>>iz<<15))
        a[iz]=1;
    else
        a[iz]=0;
    }
    fprintf(fp1,"%c",p);
    x0=x1;
    y0=y1;
    z0=z1;
    ps=0;
    pt=0;
    pk=0;
    }
    fclose(fp1);return 0;}
```

EK B RSA Genel Anahtar Sistemi

RSA Public Key Sistemi	***Örnek***
Hem p'nin hem de q'nun asal olduğu p ve q seçilir.	P =11; q=13;
Mod alınacak değer hesaplanır **n** = **p*q**.	**N** = 11*13 = 143.
Euler's totient fonksiyonu uygulanır **t** = (**p**-1)(**q**-1).	**t** = (11-1)*(13-1) = 120.
T değeri ile en büyük ortak böleni 1 olan bir e değeri hesaplanır.	**e** = 7. (7<120, ve 7 ve 120 nin en büyük ortak böleni 1 dir
e*d = 1 mod **t** olacak şekilde d değeri hesaplanır	7***d** = 1 mod 120 => **d** = 103, çünkü 7*103 = 721 = 1 mod 120.
Public key (**e**, **n**).	public key (7, 143).
Private key (**d**, **n**).	private key (103, 143).
Plaintext **M** olsun.	**M** = 5 kabul edelim.
Ciphertext **C** = **M^e** mod **n**.	Ciphertext: **C** = 5^7 mod 143= 47
Şifre çözme işlemi => plaintext = **C^d** mod **n** = (**M^e**)^**d** mod **n** = **M**.	Plaintext: 47^103 mod 143 = 5 47^103 = (5^7)^103 = 5^721 = 5*[5^720] =5*[(5^120)^6] = 5*[1^6] = 5. 5^120 = 5^t = 1 mod 143 (Euler teoremi) veya daha basitçe, **x^(e*d) = x**; bu sebepten, 5^721 = 5'tir.