KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

KAOTİK ORTAMLARDA GÜVENLİ VERİ TRANSFERİ

YÜKSEK LİSANS TEZİ

Bilgisayar Müh. Nurhan YAVUZ

TEMMUZ 2006
TRABZON

KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI

KAOTİK ORTAMLARDA GÜVENLİ VERİ TRANSFERİ

Bilgisayar Müh. Nurhan YAVUZ

Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsünce
"Bilgisayar Yüksek Mühendisi"
Unvanı Verilmesi İçin Kabul Edilen Tezdir.

Tezin Enstitüye Verildiği Tarih : 07/06/2006
Tezin Savunma Tarihi : 03/07/2006

Tez Danışmanı : Prof. Dr. Rıfat YAZICI
Jüri Üyesi : Yrd. Doç. Dr. Hüseyin PEHLİVAN
Jüri Üyesi : Doç. Dr. Temel KAYIKÇIOĞLU

Enstitü Müdürü : Prof. Dr. Emin Zeki BAŞKENT

Trabzon 2006

ÖNSÖZ

"Kaotik Ortamlarda Güvenli Veri Transferi" adlı bu çalışma, Karadeniz Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalında Yüksek Lisans Tezi olarak hazırlanmıştır.

Çalışma süresince bilgi, görüş, öneri ve yardımlarını esirgemeyen saygıdeğer danışmanım Prof. Dr. Rıfat YAZICI'ya sonsuz teşekkürlerimi sunarım.

Ayrıca yardımlarından dolayı Öğr. Gör. Dr. Selahattin KINDIKOĞLU'na, Doç. Dr. Ercan SOLAK'a ve çalışmam boyunca manevi desteklerini eksik etmeyen sevgili arkadaşlarım Uzm. Zafer YAVUZ'a, Arş. Gör. Aykut AVCI'ya, Asiye ÖZBEK'e, E. Funda AYAR'a ve Hatice DUMAN başta olmak üzere, tüm çalışma arkadaşlarıma teşekkür ederim.

Her zaman yanımda olarak manevi desteklerini eksik etmeyen aileme sonsuz saygı, sevgi ve teşekkürlerimi sunarım.

Nurhan YAVUZ

Trabzon 2006

İÇİNDEKİLER

Sayfa No

ÖNSÖZ II

İÇİNDEKİLER III

ÖZET VI

SUMMARY VII

ŞEKİLLER DİZİNİ VIII

TABLOLAR DİZİNİ X

SEMBOLLER DİZİNİ XI

1. GENEL BİLGİLER 1

1.1. Giriş 1

1.2. Bilgi Güvenliği 2

1.3. Kriptografinin Tarihi 3

1.4. Kriptolojide Kullanılan Terimler ve Kriptografinin Hedefleri 5

1.5. Şifreleme Tekniklerinin Sınıflandırılması 7

1.5.1. Algoritması Gizli Olan Şifreleme Teknikleri 8

1.5.2. Algoritması Açık Olan Şifreleme Teknikleri 8

1.6. Simetrik Şifreleme 9

1.6.1. Simetrik Anahtar Kullanarak Data Şifreleme 10

1.6.2. Blok Şifreleyiciler 11

1.6.3. Akış Şifreleyiciler 11

1.6.4. DES 12

1.7. Asimetrik Şifreleme 17

1.7.1. Açık Anahtar Kullanarak Şifreleme 18
1.7.2. Asimetrik Şifreleme İçin Gereklilikler 20
1.7.3. Sayısal İmza 21
1.7.3.1. Sayısal İmzanın Özellikleri 22
1.7.3.2. İletinin İmzalanması 22
1.7.3.3. İletinin İmzasının Doğrulanması 23
1.7.3.4. Sayısal İmzanın İnkar Edilemez Özelliği 24
1.8. Kaos ve Kriptoloji 25
1.8.1. Kaos Nedir? 25
1.8.2. Kaosun Ortaya Çıkışı 25
1.8.3. Kaotik Özellikler 28
1.8.3.1. İterasyon 28
1.8.3.2. Dallanma Diyagramı 31
1.8.3.3. Başlangıç Koşullarına Duyarlılık 32
1.8.3.4. Lyapunov Üsteli 34
1.8.4. Kaotik Sistemlerin Senkronizasyonu 35
1.8.4.1. Rössler Sisteminde Senkronizasyon 37
1.8.4.2. Lorenz Sisteminde Senkronizasyon 38
1.9. Kaos Tabanlı Kriptografi 39
1.9.1. Kaos Tabanlı Kripto Sistemlerin Gerçekleştirilmesi 41
1.9.1.1. Kaotik Kripto Sistemlerin Gerçekleştirilmesi 41
1.9.1.2. Kaotik Kripto Sistemlerin Avantajları ve Dezavantajları 43
1.9.1.2.1. Kaotik Kripto Sistemlerin Avantajları 44
1.9.1.2.2. Kaotik Sistemlerin Şifrelemedeki Dezavantajları 44

2. YAPILAN ÇALIŞMALAR 46
2.1. Lyapunov Üsteli Belirleme ve Kaotik Yörünge Üretimi 46
2.2. Yörünge Seçimi 52
2.3. Şifreleme 52
2.4. Deşifreleme 59
3. BULGULAR VE TARTIŞMA 64
3.1. Kaotik Davranışın Araştırılması 64
3.2. Oluşan Dosya Özelliklerinin Karşılaştırılması 68
3.3. Şifreleme Yöntemlerine Göre İşlem Süreleri ve Dosya Boyutları 70
4. SONUÇLAR 71
5. ÖNERİLER 72
6. KAYNAKLAR 73
ÖZGEÇMİŞ 75

ÖZET

Bilgi, geçmişten beri değerli görülmüş, uğruna rekabet edilmiş ve günümüzde de, teknolojinin gelişmesiyle birlikte en büyük güç haline gelmiştir. Artık bilgi, üretilen, iletilen, pazarlanan bir konumdadır. Sadece özel kişilerle paylaşılmak istenen bilginin bile, herhangi bir ticari, ekonomik, politik değeri olmasa dahi, farklı kişiler tarafından bilinmesi arzu edilen bir durum değildir. Bilgi, eğer, bir şirketin özel bilgilerini, devlet politikalarını, askeri stratejileri vs. içeriyorsa önem arz etmektedir ve korunmak zorundadır.

Günümüz için oldukça ilkel sayılabilecek tekniklerden, büyük matematiksel analizler gerektiren tekniklere kadar birçok yöntem kullanılarak, yüzyıllardır bilginin güvenliği sağlanmaya çalışılmıştır. Bilgisayarların ortaya çıkması ve internetin gelişimiyle birlikte de, bu güvenlik gereksinimi üst düzeye çıkmıştır. Çünkü gelişen teknoloji, beraberinde yüksek tehdit ve tehlikeler getirmiştir.

Bilgi güvenliğini sağlama üzerine gerçekleştirilen bu çalışmada, ilk olarak kriptografinin ortaya çıkışından günümüze kadar yaşanan gelişmelerden bahsedilmiş ve temel kriptografik bilgiler verilmiştir. Kaos bilimi detaylı bir şekilde incelenmiş, kaos ve kriptografinin benzer yanları irdelenmiş, kaosun kriptografiye uygulanmasının yaratacağı etkiler değerlendirilmiş ve kaotik sistemlerin genel özelliğinden de yararlanılarak verilerin şifrelenmesi ve deşifrelenmesi gerçekleştirilmiştir.

Anahtar Sözcükler: Kaos, Lyapunov Üsteli, Dallanma Diyagramı, Lojistik Harita, RC4 Algoritması

SUMMARY

Reliable Data Transfer In Chaotic Systems

Being a subject of the competition between companies and organizations, the information has always been valuable, and nowadays it has also been one of the most powerful means along with the improvement of the technology. After all, the information produced is in a form that could be delivered and marketed. Even the information that has no commercial, political, or economical value and is shared by closely related people is wanted to be private, and those people also don't want that information to be revealed. Private information about companies, government political or military strategies etc are extremely important in many respects and it has to be secure.

Over the centuries, many methods, varying from techniques that could be considered trivial comparing with todays` to the methods that requires complex mathematical analyses, have been used to try to secure the possessed information. Along with the invention of computers and the development of the internet, the necessity of security increased to the top level. It is because the developing technology also brought the threats and dangers.

In this thesis, which is about ensuring the security of information, the developments on cryptography from the beginning up to today are mentioned and the fundamental issues about the cryptography are presented at the first place. Then, chaos theory is examined in details, and the similarities between chaos and cryptography are investigated. Later, the effects of applying chaos theory on cryptography are evaluated. Finally, encryption and decryption of data is realized by utilizing general characteristics of chaotic systems.

Keywords: Chaos, Lyapunov Exponent, Bifurcation Diagram, Logistic Map, RC4 Algorithm

ŞEKİLLER DİZİNİ

Sayfa No

Şekil 1. Simetrik şifreleme 10

Şekil 2. DES'in genel yapısı 13

Şekil 3. Açık anahtarlı şifreleme-Senaryo 1 18

Şekil 4. Açık anahtarlı şifreleme-Senaryo 2 19

Şekil 5. Açık anahtarlı şifreleme-Senaryo 3 (Şifreleme Kısmı) 19

Şekil 6. Açık anahtarlı şifreleme-Senaryo 3 (Deşifreleme Kısmı) 20

Şekil 7. İletinin sayısal imzalanması 23

Şekil 8. Sayısal imzanın doğrulanması 24

Şekil 9. Lorenz deneyinde başlangıç koşullarına duyarlılık 26

Şekil 10. Lojistik haritaya ait dallanma diyagramı, r=[3,4] 31

Şekil 11. Lojistik haritaya ait dallanma diyagramı, r=[3.8,3.9] 31

Şekil 12. Lojistik haritaya ait dallanma diyagramı, r=[3.84,3.86] 32

Şekil 13. Lojistik haritada iterasyon sonucu oluşan değerler 33

Şekil 14. Başlangıç değerleri arasında 0.001'lik farkın sonucu 33

Şekil 15. Başlangıç değerleri arasında 0.000001'lik farkın sonucu 34

Şekil 16. Rössler süren sistemi için ve ($x' - z'$) sürülen sistemi ve y(t) süren işaretinin oluşturduğu çekiciler 37

Şekil 17. Lyapunov üsteli hesaplanmasına ilişkin akış diyagramı 47

Şekil 18. Çekicinin S adet bölgeye bölünmesi 48

Şekil 19. Lojistik haritanın, r=3.78, N=65536 x_n= 0.432031250 şartlarıyla oluşan sabit yoğunluk değerleri 51

Şekil 20. Lojistik haritanın, r=3.78, N=65536, x_n= 0.648565759 şartlarıyla oluşan sabit yoğunluk değerleri 51

Şekil 21. RC4 algoritmasının akış diyagramı.....55

Şekil 22. Şifreleme Algoritması 56

Şekil 23. Sınır değerlerinin aşılmaması durumu 57

Şekil 24. Sınır değerlerinin aşılması durumu 58

Şekil 25. B kontrol değerinin dosyaya yazılması 59

Şekil 26. Taşma olmaması durumunda okuma.....60

Şekil 27. Taşma durumunda dosyadan bayt okuma 61

Şekil 28. Varsa, B kontrol baytının okunması.....61

Şekil 29. Deşifreleme işlemi 62

Şekil 30. Lojistik haritaya ait dallanma diyagramı (r=[0,4]).....64

Şekil 31. Lojistik haritaya ait dallanma diyagramı (r=[3.8,3.9]).....64

Şekil 32. Lojistik haritanın r=[0,3.57] durumundaki Lyapunov üstelleri 67

Şekil 33. Lojistik haritanın r=[3.57,4] durumundaki Lyapunov üstelleri 67

Şekil 34. Lojistik haritanın r=[3.57,4] durumundaki negatif Lyapunov üstelleri 68

Şekil 35. Herhangi bir dosyanın harf yoğunluğu.....69

Şekil 36. Şifrelemiş dosyanın yoğunluğu.....69

Şekil 37. RC4 kullanılarak şifrelenmiş dosyanın yoğunluğu 70

TABLOLAR DİZİNİ

Sayfa No

Tablo 1. Başlangıç permütasyonu....12

Tablo 2. Başlangıç permütasyonunun tersi....14

Tablo 3. Anahtar permütasyonu....14

Tablo 4. Kaydırılan bit sayıları....14

Tablo 5. Sıkıştırma permütasyonu....14

Tablo 6. Genişletme permütasyonu....15

Tablo 7. P kutuları....15

Tablo 8. S-Kutuları....16

Tablo 9. $f:x \rightarrow x^2 + ¼$ fonksiyonu için farklı çekirdek değerleriyle oluşan yörüngeler....29

Tablo 10. $f:x \rightarrow x^2$-3/4 fonksiyonunun farklı çekirdek değerleriyle oluşan yörüngeler....30

Tablo 11. Rössler sisteminin alt sistemleri, süren işaretleri ve Lyapunov üstelleri....38

Tablo 12. Lorenz sisteminin alt sistemleri, süren işaretleri ve Lyapunov üstelleri....38

Tablo 13. Kaos ve kriptografi arasındaki benzerlikler....39

Tablo 14. Şifrelenecek karakterlerin, lojistik harita uzayında eşleştirildikleri aralıklar....53

Tablo 15. En küçük yörünge seçilmesi durumunda alınan sonuçlar....54

Tablo 16. Yörüngelerden herhangi biri seçilmesi durumunda alınan sonuçlar....54

Tablo 17. r=3.83 için iterasyon değerleri....65

Tablo 18. r=3.8515 için iterasyon değerleri....66

Tablo 19. r=3.87 için iterasyon değerleri....66

Tablo 20. Normal kaotik şifreleme....70

Tablo 21. RC4 sözde rasgele sayı üreteci kullanılarak yapılan kaotik şifreleme....70

SEMBOLLER DİZİNİ

a,b,c	Rössler sistemindeki katsayılar
ASCII	Alfasayısal kod
B[C_i]	C_i'nin aynı maske değeri sayısı
c	Sabit değer
C	Şifreli metin
C_i	i. karakterin şifreli hali
$\tilde{C}_i$	Maskelenmiş C_i değeri
D	Deşifreleme Algoritması
DES	Veri şifreleme algoritması
E	Şifreleme Algoritması
E(t)	t anında iki yörüngenin aralarındaki fark
$f^n(x)$	x kök değeri ile başlayan n. iterasyon
$f_n(x)$	Sözde yalancı rastgele sayı üreteci
K	Anahtar
K_n	n. anahtar
L(c)	c başlangıç koşullu yörüngenin Lyapunov üsteli
M	Açık metin
N	İterasyon sayısı
N_0	Kaotik sistemin geçiş zamanı
r	Kontrol parametresi
RC4	Rivest akış şifreleme algoritması
S	Çekicide kullanılacak bölge sayısı
S_ε	Çekicinin ε. aralığı
$\dot{u}$	Dinamik sistem
$u_1, u_2, .. u_n$	Dinamik sistemin boyutları
$\dot{v}$, $\dot{w}$	Dinamik sistemin alt sistemleri
$\dot{w}'$	Kopyalanmış alt sistem

x	Kök değeri
x_1	İterasyon başlangıç noktası
x_{min}, x_{max}	Çekicinin kullanılan kısmının sınırları
x_n	Yörüngenin n. iterasyondaki değeri
$\Delta(x_0,t)$	t anında iki yörünge arasındaki fark
ε	Çekicinin karakter aralık uzunluğu
λ	Lyapunov üsteli
σ, ρ, β	Lorenz sistemindeki katsayılar

1. GENEL BİLGİLER

1.1. Giriş

Kriptografi (Cryptography, şifreyazım), dar ve ilkel bir tanımlama ile, Yunanca'dan gelen kripto (saklı, gizli) ve graphy (yazım, yazmak) kelimelerinden türemiş bir sözcük olup, çeşitli metotlarla dijital verilerin güvenliğini ve gizliliğini sağlamayı hedeflemiş kriptoloji'nin bir dalıdır. Yaklaşık 4000 yıl önce Mısırlıların kısıtlı uygulamalarından, 20. yüzyıldaki dünya savaşlarına kadar birçok yerde kullanılmıştır. Şifreleme sanatının üstün özelliği, ordu, diplomatik servisler ve hükümet uygulamaları ile birleştirilerek, ülke sır ve stratejilerini korumak için kullanılan bir araç haline gelmiştir.

Ticari işlerde, devlet işlerinde, askeri işlerde, personel ilişkilerinde güvenli iş çalışması yapmak büyük bir sorundur. İletişimde, açık bir haberleşme kanalı kullanılıyorsa gizli tutulmak istenen bilginin yetkisiz bir kişi tarafından dinlenebileceği veya haberleşme kanalına girip veriyi bozabileceği ya da değiştirebileceği düşüncesi her zaman için önemli bir problem oluşturur. Sistemler arası bağlantılarda ya da herhangi iki nokta arasındaki haberleşmede verinin güvenli bir şekilde karşı tarafa iletildiğinden emin olunmalıdır. Bunun sağlanması gönderilen verinin şifrelenmesi ile olur. Böylece açık haberleşme kanalları kullanılarak verinin güvenli bir şekilde karşı tarafa ulaştırılması sağlanır.

Şifreleme teknolojileri, verileri korumak için temel olarak matematiksel yapıda çalışırlar. Haberleşmeyi koruma altına alır ve verilerin bulunduğu bölgelere izinsiz girişi engellerler. Diğer şifreleme teknikleri aynı zamanda birer doğrulama ve dijital imza metotlarını bir arada kullanarak bilgisayarı ve bilgileri istenmeyen ziyaretçilerden ya da mesajlardan korurlar. Günümüze kadar birçok şifreleme tekniği geliştirilmiştir. Bunlardan biri de kaotik şifrelemedir.

1892 yılında ilk kez Henri Poincaré tarafından ortaya atılan ve nedensel sistemlerin farklı başlangıç noktaları ile çok farklı sonuçlara gitmesi davranışını gösteren kaosun, kriptografinin bazı özellikleri ile oldukça benzer yönlerinin bulunması sonucu, kriptolojide de kullanılabileceği kanısına varılmıştır. Çalışmalar, yüz yıl öncesinden başlamış olmasına rağmen, 90'lı yıllardan itibaren, kaotik sistemlerin senkron davranış göstereceğini kanıtlayan çalışmalar sonrasında hız kazanmıştır.

Bu çalışmadaki amaç, kaotik özellik gösteren sistemlerin bu özelliğinden yaralanılarak, verilerin şifrelenme ve deşifrelenmesini gerçekleştirmektir. Bu amaçla, lojistik harita incelenmiş, kaotik davranış gösteren bölümlerinden yararlanılarak şifreleme, işlemlerin tersi uygulanarak da deşifreleme işlemi gerçekleştirilmiştir.

1.2. Bilgi Güvenliği

Bilgi, diğer önemli ticari varlıklar gibi, bir işletme için değeri olan ve bu nedenle korunması gereken bir varlıktır. Bilgi güvenliği bilgiyi, ticari sürekliliği sağlamak, ticari kayıpları en aza indirmek ve ticari fırsatların ve yatırımların dönüşünü en üst seviyeye çıkartmak için geniş tehlike ve tehdit alanlarından korur. Bilgi birçok biçimde olabilir. Ancak hangi biçimde olursa olsun, uygun bir şekilde korunmalıdır. Bilgi güvenliği, bu standartta güvenilirlik, veri bütünlüğü, kimlik doğrulamanın korunması olarak algılanır.

Bilgi güvenliği, politikalar, uygulamalar, yöntemler, örgütsel yapılar ve yazılım fonksiyonları gibi bir dizi uygun denetimi gerçekleştirme aracılığıyla sağlanır. Bu denetimler işletmenin belirli güvenlik hedeflerinin karşılandığını garanti altına almak için kullanılmalıdır.

Bilgi ve destek süreçleri, sistemler ve bilgisayar ağları önemli olan ticari varlıklardır. Bilginin gizliliği, güvenilirliği ve elverişliliği; rekabet gücünü, nakit akışını, karlılığını, yasal yükümlülükleri ve ticari imajı korumak ve sürdürmek için zorunlu ve gerekli olabilir. İşletmeler giderek sahip oldukları bilgi sistemleri ve ağları bilgisayar destekli sahtekarlık, casusluk, sabotaj, yıkıcılık, yangın ve sel gibi çok geniş kaynaklardan gelen tehdit ve tehlikelerle karşı karşıyadırlar. Bilgisayar virüsleri, bilgisayar korsanları ve hizmet saldırıları gibi yıkıcı kaynaklar daha yaygın, daha hırslı ve daha karmaşık hale gelmeye başlamıştır. Bilgi sistemlerine ve hizmetlerine bağımlılık, işletmelerin güvenlik tehditlerine karşı daha savunmasız olduğu anlamına gelmektedir. Genel ve özel ağların birbirleriyle bağlantısı ve bilgi kaynaklarının paylaşımı, erişim denetimini oluşturmadaki zorlukları artırmaktadır. Dağıtılmış bilgi işlemeye olan eğilim, merkezi, uzman denetimin etkinliğini zayıflatmıştır. Bilgi sistemleri henüz yeterli güvenlik seviyesinde tasarlanmamıştır. Teknik olanaklar aracılığıyla ulaşılabilen güvenlik sınırlıdır, uygun yönetim ve yöntemlerle desteklenmelidir. Hangi denetimlerin yer alacağının tanımlanması, özenli planlamayı ve detaylara dikkati gerektirir. Bilgi güvenliği yönetimi en az, tüm işletme çalışanlarının katılımını gerektirir. Aynı zamanda tedarikçilerin, müşterilerin ve ortakların da katılımına

gereksinim duyulur. İşletme dışında uzman tavsiyelerine gerek duyulabilir. Bilgi güvenliği denetimleri, eğer şartların ve tasarım aşamasının gereklerinde belirtilirse çok daha ucuz ve etkili olur [1].

1.3. Kriptografinin Tarihi

Heredot'un anlattıklarına göre eski Yunan'da şifreli bir mesaj gönderilmek istendiğinde, kölelerin kafa derisi üzerinde mesajlar aktarılmaktaydı. Önce bir kölenin kafası tıraş edilir, daha sonra da ilgili mesaj kafasına kazınır ve saçlarının uzaması beklenirdi. Birkaç ay sonra da köle, hedefine doğru yola çıkar ve gittiği yerde tekrar kafası tıraş edilerek mesaj okunurdu.

Artık ne köleler ne de aylar boyu beklenecek zaman vardır. Ayrıca pek zarif bir fikir olmayan bu yöntem yerine gelişen zaman içerisinde pek çok yeni yöntem keşfedilmiştir.

Tarihte resmi olarak kriptoyu kullanan ilk kişi Sezar'dır. Başka bir kıtadaki komutanlarına mesajlarını güvenle iletmek için *scytale* isimli değnek kullanmıştır. Sezar, şerit halindeki kağıdı değnek etrafına sarar ve mesajını boylamasına bu kağıt üzerine yazardı. Daha sonra kağıdın kıvrımlarını açarak düz hale getirir ve göndereceği adrese yollardı. Sonraları Roma orduları iletişimlerinde yine Sezar'ın bulduğu alfabede 3 harf kaydırma şifrelemesini kullanmışlardır. Bu yöntemde, örneğin "A" harfi yerine "D", "B" harfi yerine "E" kullanmaktaydı. Oldukça basit ve hedefine ulaşan bu yöntem o çağın şartları için yeterli olmuştur.

Gelişen zaman içerisinde değişen şifreleme yöntemleri birbirini izlemiş, kimi zaman çözülen bir şifre imparatorlukların kaderini değiştirmiştir. Örneğin 1587 yılında İngiliz Kraliçesini devirmek için adamlarıyla haberleşmede kullandığı basit değiştirme yöntemi çözülen İskoçya Kraliçesi, bu hatasını idam edilerek ödemiştir.

1. Dünya savaşında Almanların çözmemesi için bir Amerikan Telefon ve Telgraf şirketinden bir çalışan olan Gilbert Vernam tarafından hazırlanan "bir kerelik bloknot" yöntemi, savaş boyunca Amerika Birleşik Devletleri'nin mesaj güvenliğini sağlamıştır. Bu sistemde şifrelenecek metin ASCII kodundaki karakterlere dönüştürülür ve bir kez şifreyi çözmede kullanılacak gizli anahtar, mesajı okuyan kişi tarafından imha edilirdi. Böylece tek seferlik mesajlaşmalar, güvenli bir iletişimi oluştururdu.

2. Dünya savaşında ise filmlere konu olan Enigma makinesi Almanların en güvendiği şifreleme tekniğiydi, ta ki; Ruslara esir düşen bir Alman savaş gemisinde ele geçirilen

Enigma makinesinin İngiliz şifre kırıcılar tarafından çözülmesine kadar. Bu olay, savaşın kaderini değiştirmiştir. Almanların tüm haberleşmesini dinleyen İngilizler, bu bilgi ile uzun süre Almanların ne yapacaklarını erkenden öğrenip ona göre taktik hazırlama şansına sahip olmuşlardır.

Enigma makinesi temel olarak; klavyesinden girilen karakterlerin makine içerisinde birbiri ile değişik şekillerde algoritma oluşturacak şekillerde yazıları kodlayan üç adet diskten oluşmaktaydı. Enigma'daki diskler Almanlar tarafından önce 5'e ve daha sonra da 8'e çıkarılmıştır. Ancak bütün bu tedbirler İngilizlerin ilk bilgisayarların atalarından olan, IBM bilgisayar sistemi ile kodları çözmesini engelleyemedi.

Enigma'nın şifresinin bilgisayarlarla sonraki zamanlarda bilgisayarların şifreleme işlemlerinde daha çok kullanılmasına ve günümüzde de vazgeçilmez bir parçası olmasına neden olmuştur [2].

1960'lı yıllarda bilgisayar ve haberleşme sistemlerinin ortaya çıkması, beraberinde, bilginin dijital ortamda güvenli bir şekilde saklanması gereğini ortaya çıkarmıştır. 1970'lerin başında IBM'de başlatılıp, 1977'de Amerika Birleşik Devletleri Federal Bilgi İşleme Standardı olarak benimsenmesiyle birlikte son hali verilen DES(Data Şifreleme Standardı), tarihte bilinen en iyi şifreleme mekanizmasıdır.

Şifrelemede en büyük gelişme, 1976 yılında, Diffie ve Hellman adlı bilim adamlarının, açık anahtarlı şifreleme tekniğini ortaya attıkları, 'Şifrelemede Yeni Yönler' isimli makaleleri ile yaşanmıştır. Yazarlarının, bu makalenin yayınlanması sırasında henüz pratik olarak gerçekleştirilmiş bir uygulamalarının olmamasına karşın, fikir oldukça netti ve şifreleme ile ilgilenen bilim adamlarında yeterli ilgiyi uyandırmıştır. 1978 yılında RSA şifreleme tekniğinin kurucuları olan Rivest, Shamir ve Adleman isimli bilim adamları, ilk pratik açık anahtarlı şifreleme tekniğini gerçekleştirmişlerdir. Bu şifreleme tekniği ile birlikte, zor matematik uygulamaları şifrelemede kullanılmaya başlamıştır. Bununla birlikte, daha etkin matematiksel metotların bulunması arayışı başlanmış ve 1980'lerde bu alanda oldukça büyük gelişmeler yaşanmıştır.

Açık anahtarlı şifrelemenin en önemli katkılarından biri de, dijital imzayı sunmasıdır. 1991 yılında, dijital imza için ilk uluslararası ISO/IEC9796 standardı kabul edilmiştir. Bu standart, RSA açık anahtarlı şifreleme tekniğine dayalıdır. 1994 yılında ise, Amerika Birleşik Devletleri, ElGamal açık anahtar uygulamasını esas alan, Digital Signature Standart'ını kabul etmiştir.

1.4. Kriptolojide Kullanılan Terimler ve Kriptografinin Hedefleri

Kriptoloji, kriptoanaliz ve kriptografi olmak üzere iki kısımdan oluşur. Kriptografi, veriyi güvenli hale getirme bilimi olarak tanımlanırken, kriptanaliz de, güvenli haberleşmeyi analiz edip, şifreyi kırma bilimi olarak tanımlanmaktadır. Kriptoanaliz, ktiptografide önemli rol oynamaktadır. Çünkü şifreli metni inceleyerek, açık metnin içeriğini elde etmeyi amaçlamaktadır.

Mesajı gönderen kişiye gönderici (*sender*), mesajı alan kişiye de alıcı (*receiver*) adı verilir. Gönderici, gönderdiği mesajın başkaları tarafından okunmasını veya değiştirilmesini istemediğinden dolayı, gönderdiği bilgileri gizleme yoluna başvurur. Bu işleme şifreleme (*encryption*) adı verilir. Henüz şifrelenmemiş mesaja açık metin (*plain tex-clear text*) adı verilirken, şifreleme sonucu oluşan metne de şifreli metin (*chipher text*) adı verilir. Şifreli metinden, açık metni elde etme işlemine deşifreleme (*decription*) adı verilir.

Eğer f(x) fonksiyonu y sonucunu üretirse, y ve f fonksiyonu bilindiğinde, şifrenin kırılması için gerekli tek şey, x anahtarının bulunmasıdır ki, bunun için birçok kriptoanaliz teknikleri geliştirilmiştir.

Tüm x değerleri, anahtar uzayını oluşturur. Günümüzde uygulanan algoritmalarda mevcut bir anahtar uzayı ve bu uzayda aranan anahtarı bulmayı sağlayan bir arama metodu vardır. Her bilinen algoritma kırılabilir. Fakat algoritmanın ve anahtar uzunluğunun karmaşıklığı, işlemi sağlamlaştırır. Çok büyük hesaplama gücü, mesajın kırılması ve anahtarın bulunması için geçen zamanın büyük olması algoritmayı güçlü kılar.

Kriptografik bir algoritma, şifreleme ve deşifreleme işlemlerinde kullanılan bir matematiksel fonksiyondur. Şifreleme algoritması, tüm mümkün anahtarlar ve bu sistemi çalışır yapan tüm protokoller, kripto sistemi oluştururlar.

Kriptografik uygulamalarda anahtar, bir metni şifrelemekte veya açmakta kullanılan veri parçasına (sayı, kelime veya herhangi bir sayısal veri parçası) verilen isimdir. Anahtarlar temelde, oldukça büyük sayılardır. Anahtar boyu, bit sayısı ile ifade edilir ve açık anahtarlı şifrelemede, anahtar boyu ne kadar büyürse, şifreli metnin güvenliği de o kadar iyi olur.

Açık anahtarla özel anahtar birbiriyle ilişkilidir ancak, açık anahtardan özel anahtarın elde edilmesi çok zordur. Bununla birlikte, özel anahtar, yeterli zaman ve hesaplama gücü verildiğinde elde edilebilir. Durum böyle olunca da, anahtarın boyutunu belirleme oldukça

önem kazanmaktadır. Oldukça büyük olması güvenliği sağlarken, küçük bir anahtar, kolayca uygulamalarda işleme sokulabilir. Buna ek olarak, kimlerin dosyaya erişmek isteyeceği, bunların nasıl belirleneceği, ne kadar süreye ihtiyacı olacakları ve kaynaklarının neler olacağı da algoritma ve anahtarın yanında düşünülmesi gereken diğer unsurlardır. Büyük anahtarlar, kriptografik olarak uzun süre güvenli olabilir. Yıllar boyu gizli kalacak şifreli metinler için, çok büyük anahtar kullanmak gereklidir. Ayrıca yarının daha efektif bilgisayarları ile bu şifrenin, ne kadar bir sürede kırılacağını bugünden kestirmek zordur. Çünkü 56 bitlik bir simetrik anahtarın oldukça güvenilir olduğunun düşünüldüğü dönemler de olmuştur.

Kriptografi, sadece bilginin güvenliğini sağlamak anlamına gelmez. Bunun yanı sıra bir takım konularla da ilgilenir.

- **Gizlilik (*Confidentiality*)**

İki merkez arasında gönderilen verinin üçüncü kişiler tarafından okunmasını engelleme durumudur. Bu basit şekilde, normal yoldan gönderilen bir mektubun, alıcı kişiye giderken yolda herhangi bir kişi tarafından okunmasını (örneğin postacı) engelleme amacı güder. Yazılan mektup açık metin şeklindedir ve herhangi bir kişi tarafından zarfın açılması halinde, gönderilmiş mektup okunabilir. Şifreleme bu açık metnin, şifrelenerek yazılması işlemini gerçekleştirir. Bu sayede mektubun yolda giderken herhangi bir kişi tarafından açılması halinde yazı açık metin olmadığı için okunması engellenecektir. Gizlilik, fiziksel ortamlarda güvenlikten, matematiksel algoritmalara kadar varan birçok yaklaşımla sağlanır ve şifrelemede simetrik ve asimetrik yöntemler kullanılır.

- **Bütünlük (*Integrity*)**

İki merkez arasında gönderilen verinin üçüncü kişiler tarafından değiştirilmesini engelleme durumudur.

Normal yoldan gönderilen mektup yine üçüncü kişiler tarafından yolda orijinal şeklin dışında başka bir şekle dönüştürülerek yolculuğuna devam ettirilebilir.

Yazılan mektup açık metin şeklindedir ve içerik okunabilmektedir. Okunabilen bu açık metin kötü niyetli kişiler tarafından yolda değiştirilerek, alıcıya farklı içerikle gönderilebilir. Şifreleme gönderilen bu düz metin üzerinde işlem yaparak sayısal bir sonuç

oluşturur. Bu sonuç gönderilen yazının üzerinde en ufak bir değişiklik yapıldığında, algoritma aynı olduğundan değişecektir.

Gönderici ve alıcı tarafından aynı yazı üzerinde aynı algoritmayla oluşturulan sayısal sonuçlar birbirinin aynı olmak zorundadır. Eğer sayısal sonuçlar birbirini tutmuyorsa 'gönderilen metin yolda değiştirilmiştir' şeklinde düşünülebilir. Çünkü aynı metin üzerinde yapılacak bir değişiklik aynı sayısal sonucu çıkarmayacaktır. Kullanılan algoritma sayesinde farklı metinler üzerinde aynı sayısal sonucun çıkartılması neredeyse imkansızdır (Bu sayısal sonuç oluşturma işlemi için *Hashing* yöntemi kullanılır).

- **Reddedilemezlik (*Non-Repudation*)**

Bilgiyi oluşturan ya da gönderen, daha sonra bilgiyi kendisinin oluşturduğunu veya gönderdiğini inkâr edememelidir. Bir gönderici daha sonrasında bir ileti göndermiş olduğunu yanlışlıkla reddetmemelidir.

- **Kimlik Belirleme (*Authentication*)**

İki merkez arasında gönderilen verinin, alıcı tarafında belirtilen gönderici tarafından gönderildiğinden emin olunması durumudur (*Digital Signature*). Kişiye ulaşan bir mektubun üzerinde bulunan gönderici ismi her zaman doğru olmayabilir. Kötü niyetli kişiler tarafından gönderici isimleri farklı yazılarak kişilere mektup yollanabilir (spam mailler gibi). Şifreleme, bilim olarak bu mektuplar üzerine özel imzalar (*signature*) ekleyerek, mektubu gönderen kişinin gerçekten mektubu gönderen kişi olduğundan emin olmayı sağlayabilir.

Gönderilen zaman dilimine göre özel algoritmalarla oluşturulan bu imzalar, alıcı kişi tarafından belirli yöntemlerle doğrulanabilir. Bu imza oluşturma ve doğrulama işlemi dijital imza olarak adlandırılır [3, 4].

1.5. Şifreleme Tekniklerinin Sınıflandırılması

Şifreleme teknikleri algoritmalarına, anahtar sayısına ve şifrelenecek mesajın tipine göre sınıflandırılmıştır. Sınıflandırmada, her bir algoritma türü için en yaygın olarak kullanılan ve en temel yapıya sahip birkaç algoritma örnek olarak verilmiştir.

1.5.1. Algoritması Gizli Olan Şifreleme Teknikleri

Literatürde bulunan en ilkel şifreleme algoritmaları sadece alıcı ile gönderici arasında bilinen ve birbirinin tersi olan gizli bir algoritmaya dayanmaktadır. Bu tip algoritma ilk kez Sezar tarafından generallerine mesaj göndermek için kullanılmıştır.

Şifrelemenin güvenilirliği, algoritmanın kendisi saklı kaldığı müddetçe geçerlidir. Büyük işletmeler için uygun bir şifreleme yöntemi değildir. Kullanılan algoritmayı bilen birinin işten ayrılması veya kazara algoritmanın açığa çıkması durumunda sistemin yeni bir algoritmaya göre yeniden güvenliğinin sağlanması gereklidir. Bu da işletmenin büyüklüğüyle orantılı olarak artan, başlı başına hacimli bir iştir. Bu nedenle işletmeler güvenliğin sağlanmasında anahtar tabanlı algoritmaları tercih etmektedirler.

1.5.2. Algoritması Açık Olan Şifreleme Teknikleri

Gizli algoritmaya dayalı şifreleme sistemleri, algoritmanın gizliliğinin sağlanmasını zorunlu kılması nedeniyle, sadece sınırlı bir kullanım alanına sahiptir ve standart hale getirilmesi mümkün değildir. Özellikle, bankalar ve elektronik ticaret siteleri gibi yaygın iletişim ağlarına sahip kuruluşlar için gizli algoritmaya dayalı şifreleme sistemleri uygun değildir. Ayrıca, şifreleme sisteminin güncellenmesi gerektiğinde gizli algoritmaya dayalı şifreleme sistemleri esnek bir yapıya sahip olmadıklarından eski sistemin tamamen kaldırılıp yerine yenisinin kurulması gerekir [5, 24, 25].

Şifreleme algoritmalarının geniş bir kullanım alanına sahip olabilmesi için, standart hale getirilmesi gerekir. Bu standart hale getirme ve güncelleme gereksinimini karşılamak için algoritması herkes tarafından bilinen şifreleme sistemleri geliştirilmiştir. Saklı olan şifreleme için kullanılan anahtarın kendisidir. Algoritma açıkça bilinse de anahtar gizli olduğu için algoritma çıktısı (şifrelenmiş veri) gizlenmiş olur. Bu tür şifreleme sistemlerine, açık metin sadece gönderen ile alıcı tarafından bilinen bir anahtarla şifrelendiği için bir anahtara dayalı şifreleme sistemi de denir. Modern şifreleme tekniklerinin büyük çoğunluğu açık algoritmaya dayalı teknikler kullanır.

Algoritması açık olan şifreleme tekniğinde, bir veri şifreleneceği zaman, şifreleme anahtarı kullanılır. Şifre çözüleceği zaman ise karşılık gelen şifre çözücü anahtar kullanılır. Bu anahtarın gizli tutulması son derece kritiktir çünkü bu anahtarı eline geçiren herhangi biri bütün mesajları çözebilecektir. Esasen şifreleme ve şifre çözme işlemleri oldukça

kolaydır. Zor olan ise anahtarların güvenli bir şekilde saklanıp, gerekli olduğunda yine güvenli bir şekilde ilgili şahıslara gönderilmesidir.

Algoritması açık olan şifreleme tekniği kullanan sistemler, anahtar sayısına göre simetrik ve asimetrik şifreleme teknikleri olmak üzere ikiye ayrılır: Simetrik ve asimetrik şifreleme.

1.6. Simetrik Şifreleme

Geleneksel veya özel anahtarlı şifreleme olarak da adlandırılan simetrik şifrelemede, şifreleme ve deşifreleme için tek bir anahtar kullanılır. Gönderen taraf, mesajı bir anahtarla şifrelerken, alıcı taraf da aynı anahtarı kullanarak şifreyi deşifreler.

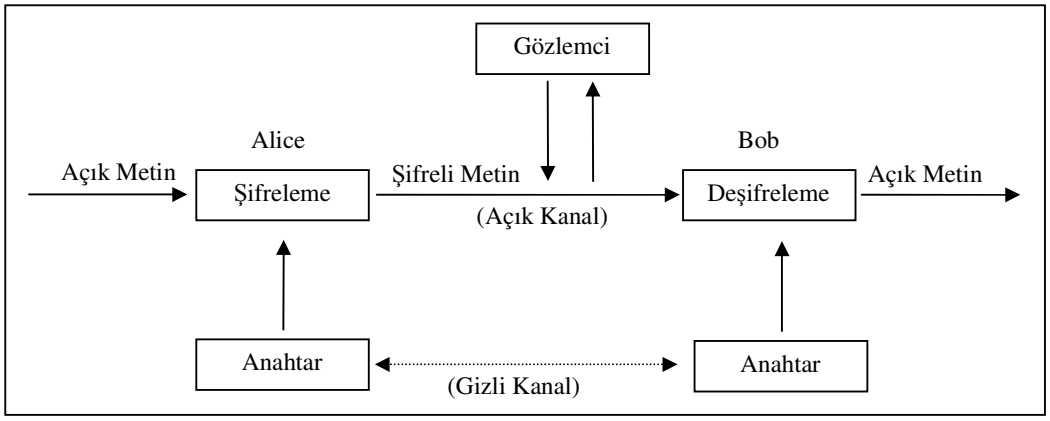
Alıcı ve göndericinin simetrik şifreleme kullanarak güvenli bir şekilde haberleşmesi için, bir anahtar üzerinde anlaşmaları ve bu anahtarı gizli tutmaları gerekmektedir. Eğer bu kişiler ayrı konumlarda bulunuyorsa, taşıyıcının, telefon sisteminin ya da diğer taşıma ortamlarının özel anahtarın saklanabilmesi açısından yeterli güvenilirlikte olması gerekmektedir. Çünkü anahtarı ele geçirecek her kişi, şifreyi çözebilir. Anahtarların üretimi, iletimi ve saklanması anahtar yönetimi olarak adlandırılır ve tüm şifreleme sistemleri anahtar yönetimi sorunlarıyla uğraşmak durumundadır. Anahtarların gizli kalmasını gerektirdiğinden dolayı, simetrik şifreleme, özel anahtar yönetiminde oldukça sıkıntı yaşamaktadır.

DES, Blowfish, Twofish, AES, CAST128, RC5 simetrik şifreleme algoritmalardır. Bu algoritmaların en büyük avantajı basit ve kolay uygulanabilir olmasıdır. Bununla birlikte daha hızlı ve verimlidirler. Ancak, şifreleme ve şifre çözme için aynı anahtarın kullanılıyor olması dezavantaj doğurur. Tek bir anahtarın güvenliğinin sağlanması zordur. Diğer şahıslara bu anahtarın güvenli olarak gönderilmesi sorununun yanı sıra, bu şahısların anahtarı ne kadar gizli tutacağı sorun teşkil etmektedir. O nedenle, bu tür algoritmalar, daha çok paylaşımın olmadığı durumlar için uygundur. Bilgisayardaki dosyaların veya sabit diskin şifrelenmesi gerektiğinde kullanılabilirler.

Data şifreleme standardı olarak bilinen DES'teki en büyük problem de, anahtarın başkalarının eline geçmeden nasıl güvenli bir şekilde dağıtılacağıdır. DES, şifrelenecek bloğun iki parçaya bölünmesi ve her aşamada sadece biri üzerinde işlem yapılması esasına dayanır.

1.6.1. Simetrik Anahtar Kullanarak Data Şifreleme

Alice, Bob'a gizli bir mesaj göndermek istemektedir. Bob ile daha önce kararlaştırdıkları ve kimseyle paylaşmadıkları anahtarı kullanarak, mesajı gizler ve Bob'a gönderir. Bob, aynı anahtarı kullanır ve mesajı deşifreler. Bu mesaj trafiğini dinleyen hiç kimse, eğer ikisinin de anlaştıkları anahtarı bilmiyorsa, şifreyi çözemez.



Şekil 1. Simetrik şifreleme

Şekil 1'de simetrik bir şifreleme, kabaca ifade edilmektedir. Şifreleme işlemi, bir Caesar şifreleme tekniği kadar basit olabileceği gibi, karmaşık matematiksel fonksiyonlarla da sağlanabilir. Caesar şifreleme metodunda, alfabedeki harflerin şifreli hali, 3 harf sonrasındaki harf olmaktadır.

ABCDEFGHIJKLMNOPQRSTUVWXYZ şeklindeki bir alfabede harflerin dizilişi, her bir harfin 3 harf sağa kaydırılması sonucu DEFGHIJKLMNOPQRSTUVWXYZABC şekline dönüşmektedir. Bu şekilde, A harfinin şifreli karşılığı D harfi, B harfininki E harfi… şeklinde olmaktadır. Günümüz standartları ile kıyaslandığında oldukça zayıf olan bu algoritma, geleneksel şifrelemenin nasıl çalıştığına ilişkin basit bir örnek teşkil etmektedir.

Şifreleme işleminin matematiksel notasyonlarla ifadesi şu şekilde olur:

M : Açık Metin

C : Şifrelenmiş Metin

K : Anahtar

E : Şifreleme Algoritması

D : Deşifreleme Algoritması

E_K (M) = C (Anahtarla Şifreleme)

D_K (C) = M (Anahtarla Deşifreleme)

Buradan şu sonuca ulaşılabilir:

D_K (E_K (M)) = M

Simetrik şifreleme algoritmaları, iki gruba ayrılır: Blok ve akış şifreleyiciler

1.6.2. Blok Şifreleyiciler

Blok şifreleyici, açık metindeki sabit boyutlu veri bloğunu, şifreli metindeki aynı uzunluklu başka bir veri bloğuna dönüştürür. Bu dönüşüm, kullanıcıya bağlı bir gizli anahtar yardımıyla yapılır. Deşifreleme ise, aynı gizli anahtarı kullanarak, algoritmanın tersinin uygulanması ile sağlanır. Bloğun sabit boyutu, çoğu blok şifreleyicide 64 bit uzunluğundadır. Farklı açık metin blokları, farklı şifreli metin bloklarına map edilir. Bu nedenle, bir blok şifreleyici, tüm mümkün mesajlardan, bire bir ters çevrilebilir permütasyonlar yaratır. Permütasyon, şifrelemede gizli tutulmalıdır. Çünkü gizli anahtarın bir fonksiyonudur.

1.6.3. Akış Şifreleyiciler

Akış şifreleyici algoritmalar, blok şifreleyici algoritmalara göre çok hızlı üretilebilirler. Blok şifreleyiciler büyük veri blokları üzerinde işlem yaparken, akış şifreleyiciler açık metindeki genelde bit olan küçük veri birimleri üzerinde işlem yaparlar. Blok şifreleyici kullanarak herhangi bir boyuttaki açık metnin şifrelenmesi sonucu, aynı anahtar kullanıldığı sürece, aynı şifreli metin üretilir. Akış şifreleyici ile açık metin parçaları, işleme sokuldukları zamana bağlı olarak değişik şekilde şifrelenirler. Akış şifreleyiciler, bit dizisi şeklinde akış anahtarları üretirler ve bu anahtarlar açık metinle XOR işlemine sokularak şifreleme işlemi gerçekleştirilir. Akış anahtarının üretilmesi, açık metne ve şifreli metne bağlı olabileceği gibi, veriye ve bu verinin şifreleme tarzına da bağlı olabilir. Açık metne veya şifreli metne bağlı olarak akış anahtarı üretilen sistemler, senkron akış şifreleyici olarak, veriye ve şifreleme tarzına bağlı olarak akış anahtarı üretilen sistemler de, kendi kendine (*self*) senkron sistemler olarak adlandırılır.

1.6.4. DES

DES (Data Şifreleme Standardı) algoritması, 1970 yılında IBM tarafından geliştirilen Lucifer algoritmasının biraz daha gelişmiş halidir. 1974'te IBM'in NSA ile birlikte geliştirdiği algoritma olan DES'in yayınlanmasından itibaren DES algoritması üzerinde geniş ölçüde çalışmalar yapılmıştır [6].

İlk tasarlandığında donanım uygulamalarında kullanılması amaçlanmıştır. İletişim amaçlı kullanımda hem gönderen, hem de alıcı şifreleme ve deşifrelemede kullanılan aynı gizli anahtar üzerinde anlaşmış olmalıdır. Gizli anahtarın güvenli bir biçimde dağıtımı için açık anahtarlı sistem kullanılabilir. DES aynı zamanda, sabit diskte veri saklamak gibi tek kullanıcılı şifreleme amaçlı da kullanılabilir. DES'in en büyük zayıflığı 56 bitlik anahtarıdır. Geliştirildiği zamanlarda çok iyi bir şifreleme algoritması olmasına rağmen, modern bilgisayarlar tarafından yapılan anahtar saldırılarına karşı yetersiz kalmıştır. DES'in diğer bir zayıflığı da yavaş olmasıdır.

DES, her birinde karıştırma (*confusion*) ve yayma (*diffusion*) tekniklerinin gerçekleştirildiği 16 döngüden oluşur. Açık metin blokları 64 bittir ve başlangıçta uygulanan başlangıç permütasyonu sonrasında, bu blok 32 bitlik iki alt bloğa bölünür. Bu bloklar üzerinde 16 kez gerçekleştirilen f fonksiyonu sonrasında, alt bloklar anahtarla işleme sokulur, birleştirilir ve başlangıçta uygulanan döngünün tersi uygulanarak işlem sonlandırılır.

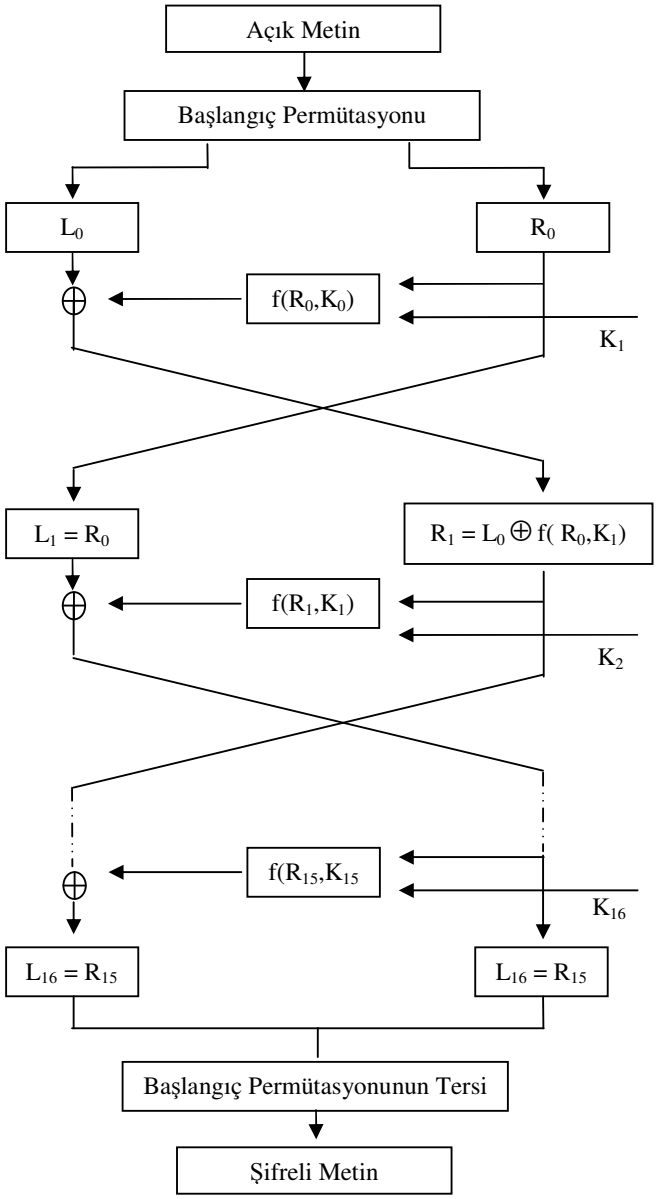
Her döngü, bir önceki döngüden gelen mesajı ikiye ayırır: L_i ve R_i , i=1,2,…,16 . İşlemler R_i üzerinde yapılır. Her döngü için anahtardan döngü anahtarı üretilir. Deşifreleme işleminde de aynı algoritma kullanılır. Ancak, anahtarların kullanım sırası tersten olur. DES'in genel yapısı Şekil 2'de gösterilmektedir.

DES' birçok aşamadan oluşmaktadır. Bunlar sırasıyla şu şekilde gerçekleşmektedir:

Başlangıç Permütasyonu: Başlangıç permütasyonu, DES'e hiçbir kuvvet katmamaktadır. Sadece bitlerin yerini değiştirir.

Tablo 1. Başlangıç permütasyonu

58	50	42	34	26	18	10	2	60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6	64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1	59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5	63	55	47	39	31	23	15	7



Şekil 2. DES'in genel yapısı

Başlangıç Permütasyonunun Tersi: Başlangıç permütasyonunun tersi olan permütasyon son döngüden sonra uygulanır. Permütasyonun detayları Tablo 2'de gösterilmektedir.

Anahtar Permütasyonu ve Döngü Anahtarının Üretilmesi: Anahtar üzerinde yapılan ilk işlem sonucu, anahtar uzunluğu 64 bitten 56 bite indirilmektedir. Bunun için her 8. bit doğruluk kontrolü için atılır. Sonra 56 bitlik anahtar, Tablo 3'teki permütasyona girer.

Tablo 2. Başlangıç permütasyonunun tersi

40	8	48	16	56	24	64	32	39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30	37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28	35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26	33	1	41	9	49	17	57	25

Tablo 3. Anahtar permütasyonu

57	49	41	33	25	17	9	1	58	50	42	34	26	18
10	2	59	51	43	35	27	19	11	3	60	52	44	36
63	55	47	39	31	23	15	7	62	54	46	38	30	22
14	6	61	53	45	37	29	21	13	5	28	20	12	4

Anahtar permütasyonu sonrasında, 56 bitlik anahtar 28 bitlik sağ ve sol olmak üzere iki parçaya ayrılır. Döndürme olarak adlandırılan kısımda, 28 bitlik parçalar her döngü için 1 ya da 2 bit sola kayar. Bu kaydırmada, kayan bitler sona eklenir. Bitlerin kayma miktarları, Tablo 4'te gösterilmektedir.

Tablo 4. Kaydırılan bit sayıları

Döngü	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Kayma	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Sıkıştırma Permütasyonu: Anahtar döngüye gönderilmeden önce, tekrar bir permütasyon gerçekleşir. Bu permütasyon sonucu 56 bitlik anahtar, 48 bite iner(Tablo 5).

Tablo 5. Sıkıştırma permütasyonu

14	17	11	24	1	5	3	28	15	6	21	10
23	19	12	4	26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40	51	45	33	48
44	49	39	56	34	53	46	42	50	36	29	32

f fonksiyonu: Her döngüde sağ 32 bitlik kısım (R_i) üzerine işlemler yapılır. Öncelikle bu 32 bitlik kısım 48 bite genişletilir. Böylece, sağ alt blok, anahtarla aynı boya yükseltilmiş olur. Bu işlem, Tablo 6' da gösterildiği şekilde gerçekleştirilir.

Tablo 6. Genişletme permütasyonu

32	1	2	3	4	5	4	5	6	7	8	9
8	9	10	11	12	13	12	13	14	15	16	17
16	17	18	19	20	21	20	21	22	23	24	25
24	25	26	27	28	29	28	29	30	31	32	1

48 bitlik bu kısım döngü anahtarı ile XOR işlemine ($\oplus$) gönderilir. Sonra 48 bit, 6 bitlik 8 gruba bölünür ve her bir grup ayrı bir S–kutusuna gönderilir. S-kutularının altışar bitlik girişleri ve dörder bitlik çıkışları vardır. S-kutuları 4 satır ve 16 sütundan oluşur. Altı bitlik girişin ilk ve son bitleri, bu tabloların satır numarasını oluştururken, ortadaki dört bitlik sayı da kullanılacak sütun bilgisini verir. DES'in güvenliği Tablo 8'de gösterilen S-kutuları ile sağlanır, çünkü bu kutularla yapılan işlemlerin analizi zordur.

S-kutuları blok şifrelerin doğrusal olmayan kısımlarıdır. Hiçbir S-kutusu, girdinin doğrusal fonksiyonu değildir. S-kutularından çıkan 4 bitlik parçalar daha sonra P-kutusuna göre permütasyona sokulur ve giriş bitlerinin belirli bir çıkış biti olarak belirlenmesi işlemi gerçekleştirilir(Tablo 7).

Tablo 7. P kutuları

16	7	20	21	29	12	28	17	1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9	19	13	30	6	22	11	4	25

Permütasyondan çıkan 32 bitlik kısım döngünün başından ayrılan 32 bitlik sol alt blokla XOR işlemine sokulur. Sol ve sağ alt bloklar yer değiştirilir ve bir sonraki döngü gerçekleştirilir.

Son olarak, başlangıç permütasyonunun tersi işlemi uygulanır. Bunun için, en son döngüde oluşan bloklar yer değiştirmez ve birleştirilerek permütasyona sokulur.

Tüm bu işlemler sonrasında şifreleme gerçekleştirilmiş olur. Deşifreleme işlemi için, şifrelemede gerçekleştirilen uygulamalar aynen gerçekleştirilir [25].

Tablo 8. S-Kutuları

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
1. S-kutusu																
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
2. S-kutusu																
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
3. S-kutusu																
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
4. S-kutusu																
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
5. S-kutusu																
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
6. S-kutusu																
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
7. S-kutusu																
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
8. S-kutusu																
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

1.7. Asimetrik Şifreleme

Simetrik şifrelemedeki anahtar paylaşım sorununu çözmek için, 1976 yılında Whitfield Diffie ve Martin Hellman tarafından asimetrik şifreleme tekniği geliştirilmiştir. Bu yöntemde, bilindiği gibi iki ayrı anahtar kullanılması ve herhangi bir anahtar transferinin gerekmemesi güvenliği artırmaktadır. Kullanılan anahtar ilgili şahıstan başka kimseyi ilgilendirmez. Anahtarların boyu ne kadar uzun olursa, şifrenin kırılma ihtimali o kadar az olur. Ancak, bu yöntem, fazla CPU işlemi gerektirmektedir ve daha çok, verinin paylaşılması veya network üzerinden dolaşması gereken durumlarda tercih edilmektedir.

Bu şifreleme türünün karakteristikleri şu şekildedir:

1- Sadece şifreleme algoritması ve deşifreleme anahtarı verilmişken, bir takım hesaplamalar yolu ile şifreleme anahtarını bulmak mümkün değildir.

2- Her iki benzer anahtar da şifreleme ve deşifreleme için kullanılabilir. Bununla beraber, bir anahtar şifreleme için kullanılmışsa, deşifreleme için diğer anahtar kullanılmalıdır [7].

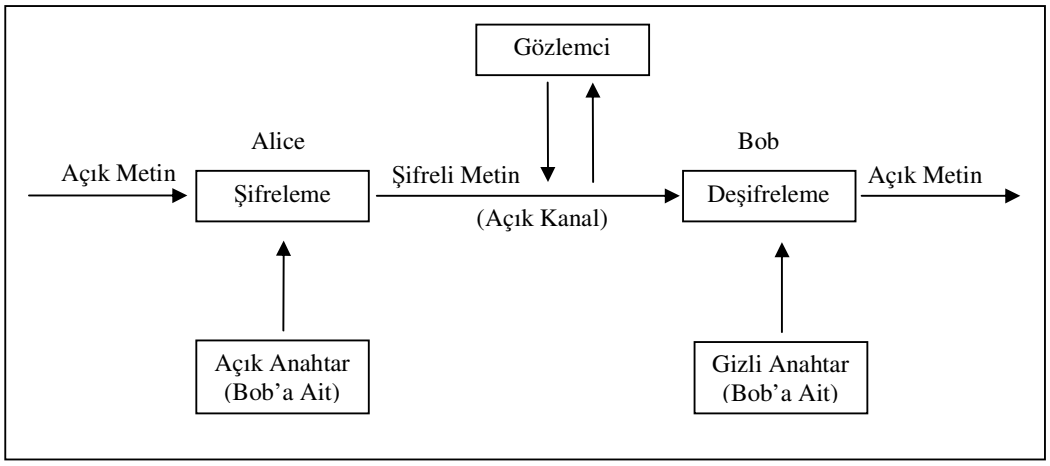
Açık anahtarlı şifreleme sistemi olarak da adlandırılan asimetrik şifreleme sistemlerinin iki ana kullanım alanı vardır: Şifreleme ve dijital imza. Bu tür sistemlerde, bir açık bir de özel anahtar olmak üzere iki anahtar kullanılır. Açık anahtar, adından da anlaşılacağı gibi herkesin bilmesinde sakınca olmayan anahtardır. Özel anahtar ise mutlaka gizli tutulmalıdır. Bu şifreleme tekniğinde alıcı ve göndericinin anahtar paylaşım gereksinimi ortadan kaldırılmıştır. Tüm haberleşmeler sadece açık anahtarla yapılırken, özel anahtarın paylaşılması veya başka bir konuma gönderilmesi gibi bir durum söz konusu değildir. Bu sistemde, haberleşme sisteminin güvenliği zorunlu değildir. Tek gerekli şey, doğrulama açısından zorunlu olan, açık anahtarların kullanıcıları ile ilişkilendirilmesinin iyi bir şekilde yapılmasıdır. Bu durum, güvenilir bir dizinde bu bilgilerin saklanması ile sağlanabilir. Herhangi biri, açık bilgiyi kullanarak gizli bir mesaj gönderebilir ancak bu mesaj, sadece mesajı deşifrelemesi gereken kişinin özel anahtarı ile deşifrelenir. Bununla birlikte, açık anahtarlı şifreleme sistemleri sadece şifrelemede değil, dijital imza olarak doğrulamada da kullanılmaktadır.

Açık anahtarlı şifrelemede, özel anahtar her zaman matematiksel olarak açık anahtara bağlıdır. Bu yüzden, açık anahtardan özel anahtar üretilerek açık anahtarlı bir sistemin şifresinin kırılması her zaman için mümkündür. Bu duruma önlem olarak, açık anahtardan özel anahtar türetilmesinin mümkün olduğunca çok zor gerçekleşmesi sağlanmalıdır.

Örneğin, bazı açık anahtarlı sistemlerde, açık anahtardan özel anahtar üretilmesi, sisteme saldıranın, çok büyük bir sayıyı çarpanlarına ayırmasını gerektirir. Bu durum da, gerçekleşmesi çok zor bir hesaplamayı doğurur. RSA, El Gamal, Knapsack, RC2, IDEA açık anahtarlı şifreleme tekniklerinden bazılarıdır.

1.7.1. Açık Anahtar Kullanarak Şifreleme

Alice, Bob'a gizli bir mesaj göndermek istemektedir. Bob'un açık anahtarını alarak, mesajını bu anahtarla gizler ve mesajı gönderir. Bob, kendisine ait olan özel anahtarı kullanır ve mesajı deşifreler. Bu mesaj trafiğini dinleyen hiç kimse, şifreyi çözemez. Bob'a herkes mesaj gönderebilir. Ancak bu mesajları sadece Bob okur. Çünkü Bob'a gönderilen mesajlar, sadece Bob'un özel anahtarı ile deşifrelenebilir.

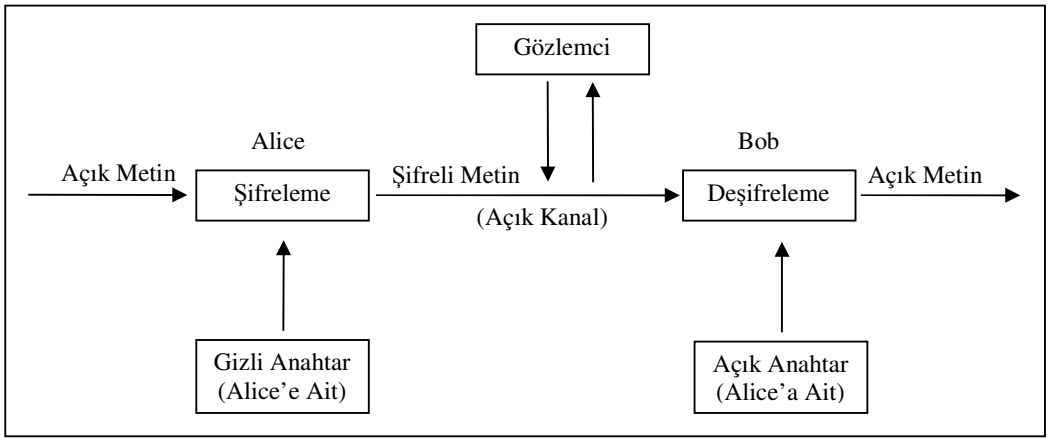


Şekil 3. Açık anahtarlı şifreleme-Senaryo 1

Bu senaryo ile Bob, sadece kendisinin okuduğundan ve başka herhangi bir kimsenin görüntüleyemediğinden emin olduğu bir mesaj alır. Fakat bunun kimden geldiğinden emin olamaz. Sadece gizlilik sağlanmış olur. Senaryo şu şekilde değiştirilmiş olsun:

Eğer Alice, Bob'a, Bob'un Alice'den geldiğine emin olarak okuyabileceği bir mesaj göndermek isterse, mesajı kendisinin gizli anahtarını kullanarak şifreler. Bob, mesajı aldığında, bu mesajı Alice'in açık anahtarı ile deşifreler. Herhangi bir üçüncü kişi de bunu yapabilir. Çünkü Alice'in açık anahtarı herkes tarafından bilinmektedir. Bu durumda Bob, bu mesajın Alice'in kendisinden geldiğinden ve kendisine ulaşana kadar yolda herhangi bir

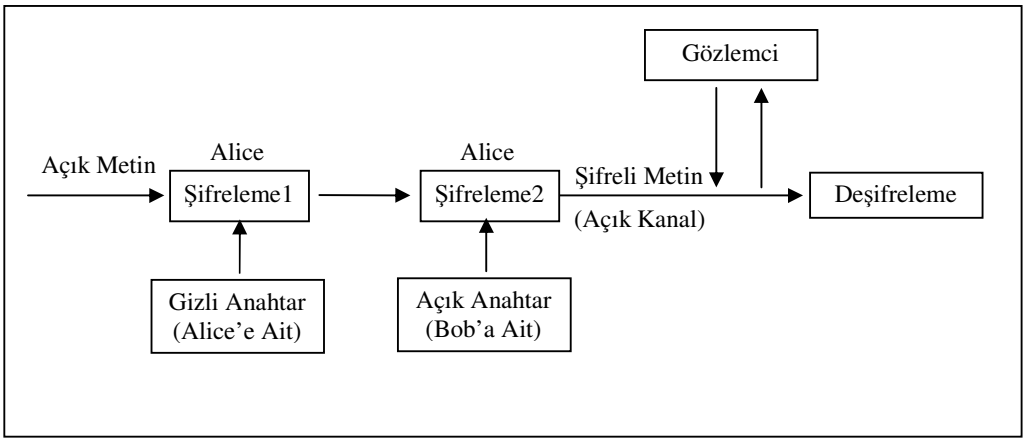
yerinin değiştirilmediğinden emin olur. Çünkü Alice'in açık anahtarı ile deşifrelediği mesajın sadece Alice'in bilebileceği özel bir anahtar ile şifrelenmiş olabileceğini bilir.



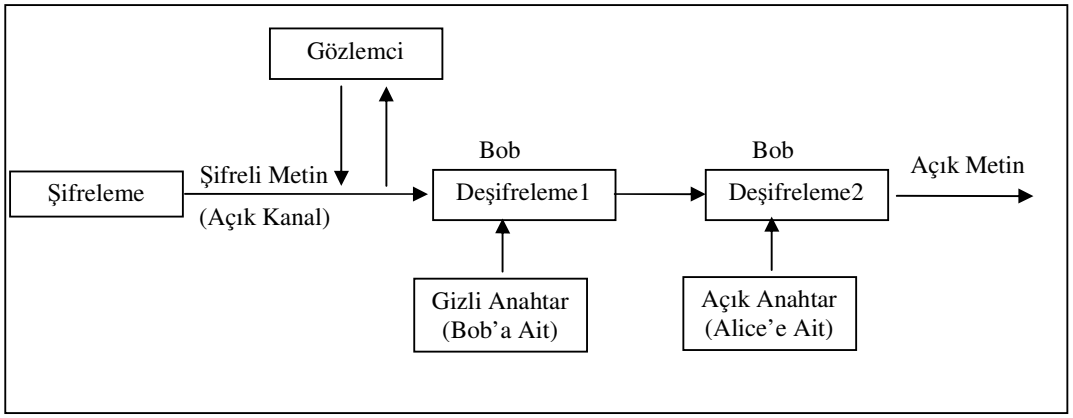
Şekil 4. Açık anahtarlı şifreleme-Senaryo 2

Bu senaryo ile de, gizlilik yerine kimlik denetimi sağlanmış olur. Hem gizliliğin hem de kimlik denetiminin sağlanabileceği bir senaryo şu şekilde olabilir:

Eğer Alice, Bob'a, Bob'un Alice'den geldiğine ve yolda kendisinden başka kimsenin içeriğini görüntüleyemediğine emin olarak okuyabileceği bir mesaj göndermek isterse, mesajı kendisinin gizli anahtarını kullanarak şifreler, daha sonra ortaya çıkan mesajı da Bob'un açık anahtarı ile şifreler.



Şekil 5. Açık anahtarlı şifreleme-Senaryo 3 (Şifreleme Kısmı)



Şekil 6. Açık anahtarlı şifreleme-Senaryo 3 (Deşifreleme Kısmı)

1.7.2. Asimetrik Şifreleme İçin Gereklilikler

1- Alıcı taraf olarak açık ve özel anahtar parçalarını oluşturmak, hesapsal olarak kolay olmalıdır.

2- Gönderenin mesajı göndereceği kişinin genel anahtarını ve şifrelenecek olan mesajı bildiği durumda, uygun şifreli metni oluşturmak, hesapsal olarak kolay olmalıdır.

$C = E_{K_1}(M)$

3- Alıcı tarafın özel anahtarını kullanarak, şifrelenmiş mesajı orijinal haline getirme yükü az olmalıdır.

$M = D_{K_2}(C) = D_{K_2}(E_{K_1}(M))$

4- Herhangi bir kişinin genel anahtarı bilerek, özel anahtarı belirlemesi, hesapsal olarak imkansız olmalıdır.

5- Herhangi bir kişinin genel anahtarı ve şifreli metni bilerek orijinal mesajı elde etmesi hesapsal olarak imkansız olmalıdır.

Bu maddelere ek olarak, yararlı olan ancak gerekli olmayan bir madde daha eklenebilir.

6- Şifreleme ve deşifreleme fonksiyonları her iki sıra ile de uygulanabilir olmalıdır.

$M = E_{K_1}(D_{K_2}(M))$

Tüm bu şartlar, sağlanması zor gerekliliklerdir. Bu nedenle, açık anahtarlı şifreleme fikrinin ileri sürüldüğünden günümüze kadar geçen yıllar süresince sadece birkaç algoritma geniş bir kitle tarafından kabul edilmiştir.

Açık anahtarlı şifrelemede kullanılan fonksiyonlar, fonksiyonun bire bir olduğu aralıkta, tersini hesaplamanın imkansız, kendini hesaplamanın ise kolay olduğu tek yönlü fonksiyon türündedirler.

$Y = f(X)$ işlemi çok kolay iken,
$X = f^{-1}(Y)$ işleminin çok zor olması durumu (1)

Burada kolay denilirken, fonksiyonun girdi uzunluğuna bağlı olarak polinominal bir zaman süresi içerisinde çözülebilir olması kastedilmektedir. Şöyle ki, eğer girdi uzunluğu n bit kadarsa, α bir sabit sayı iken, fonksiyonun hesaplanması için gereken süre, n^{α} gibi bir fonksiyonla orantılı olmalıdır. İmkansızın anlamı ise, daha bulanıktır. Giriş büyüklüğüne bağlı olarak çözüm için harcanan çaba polinominal zamandan daha hızlı artmaktadır. Örneğin fonksiyonun n bitlik bir girdisi varken, fonksiyonun çözülme zamanı 2^n gibi bir fonksiyona bağlı olarak artıyorsa, bu fonksiyonun çözümü imkansızdır denebilir.

Kriptografi için, bir taraftan hesaplanması kolay, diğer bir taraftan ise, belirli ek bilgiler bilinmedikçe hesaplanması olanaksız olan tuzak kapılı tek yönlü fonksiyonlar (*trap door one way function*) da kullanılmaktadır. Bu fonksiyonlar, tersine çevrilebilir fonksiyonların bir ailesidir [7].

$Y = f_k(X)$ k ve X biliniyorsa, kolay….
$X = f^{-1}_k(Y)$ k ve Y biliniyorsa, kolay… (2)
$X = f^{-1}_k(Y)$ Y biliniyor fakat k bilinmiyorsa çözülemez…

Uygulamalı bir açık anahtarlı kriptografi algoritmasının geliştirilmesi tuzak kapılı tek yönlü fonksiyonun bulunuşuna bağlıdır.

1.7.3. Sayısal İmza

Açık anahtarlı şifrelemenin bir diğer yararı da, sayısal imzayı sağlayacak metotlar sunmasıdır. Günlük hayatta kullanılan imzalarda olduğu gibi, sayısal imzalar da elektronik ortamda gönderilen bilginin veya e-mail'in kime ait olduğunu göstermek için kullanılır.

Açık Anahtar Altyapı (*Public Key Infrastructure*) çatısının kullanılmaya başlanması ile yaygınlaşan sayısal imza, bir anahtar çifti (açık ve özel anahtarlar) ile elektronik ortamda

iletilen veriye vurulan bir mühürdür. Karmaşık (*complex*) algoritmaların meydana getirdiği şifreleme teknolojisini kullanarak oluşturulmuş sayılar serisidir. Yani belgenin içeriğinin şifrelenerek saklanmasıdır. Sayısal imzalar göndericinin kimliğinin kesin bir biçimde teyit edilmesini ve elektronik dokümanın bütünlüğünün kontrolünü mümkün kılar. İnkar edilemez özelliktedir (*Non-repudiation*).

Sayısal bir imza, kişinin el yazısı ile attığı imzaya eş değerdir. Aynı amaçla kullanılır. Ancak, el yazısı ile atılan imzanın taklit edilmesi kolaydır. Buna karşın, sayısal imzanın taklit edilmesi nerdeyse imkânsızdır.

Sayısal imzaların oluşturulmasında ve doğrulanmasında sayısal sertifikalar kullanılır. Gönderilen verinin imzalanması için, gönderen kişiye ait bir sayısal sertifika olması gerekmektedir.

1.7.3.1. Sayısal İmzanın Özellikleri

1- Sayısal imza bir kullanıcı, sunucu ya da host'tan gönderilen bilgilerin kesinlikle o kuruma veya kişiye ait olduğunu doğrulayarak, verinin başkası tarafından yollanmadığını garanti eder.

2- Sayısal imza, veri akışı sırasında bilgilerin içeriğini korur, bir başka kişinin eline geçmesini ya da değiştirilmesini engeller, bilginin sadece alıcıya gittiğini ve sadece alıcı tarafından okunacağını garanti eder.

3- Sayısal imza, veriyi gönderenin ve alanın kim olduğunun kanıtlanmasına imkân tanır. Yani imzalanmış bir dokümanı yollayan kişi onu yolladığını inkâr edemez ve alıcı da aldığını inkâr edemez.

1.7.3.2. İletinin İmzalanması

1- İletinin mesaj özeti çıkarılır. Bunun için, uygun algoritmalar kullanılır. Mesaj özetinden orijinal iletinin elde edilmesi mümkün değildir ve orijinal iletide küçük bir değişim (bir bit'in veya karakterin değişmesi), iletinin mesaj özetinde büyük değişikliklere neden olmaktadır.

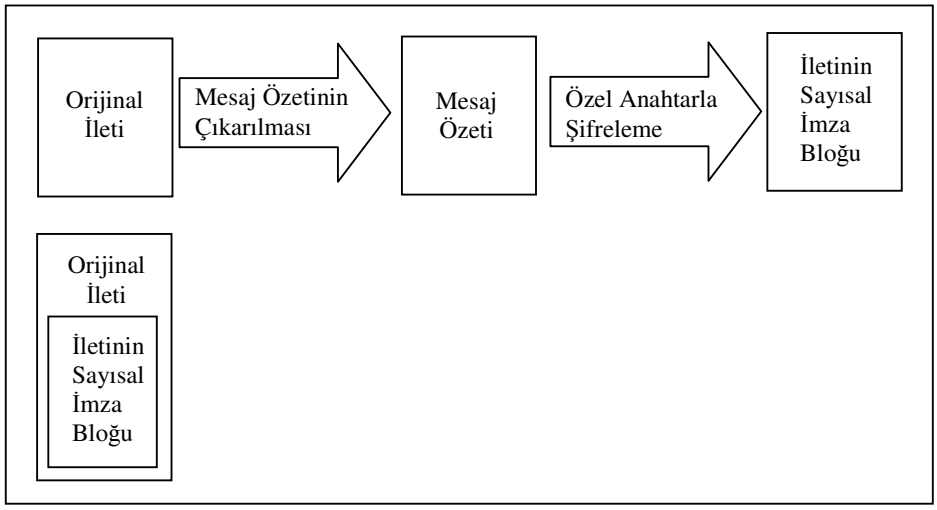
Mesaj özetinin çıkarılması için en çok SHA veya MD5 gibi algoritmalar kullanılmaktadır. MD5, Ron Rivest tarafından 1992 yılında tasarlanmış bir MD (Message Diggest) algoritmasıdır. Bu algoritma, sonsuz uzunlukta veriyi girdi olarak kabul eder ve

sonuçta 128 bit uzunluğunda bir çıktı üretir. SHA (Secure Hash Algorithm) ise, MD5'e benzeyen bir algoritmadır. MD5'le aralarında bir takım farklılıkları vardır: MD5'te çıktının uzunluğu 128 bit iken, SHA'da 160 bittir. Girdi olarak $2^{64} - 1$ uzunluğunda veriyi kabul eder. Ürettiği 160 bitlik sonuç ile kaba kuvvet ataklara karşı daha dayanıklıdır.

MD5 ve SHA'nın yanında bir çok MD algoritması tasarlanmıştır. Bunlardan bazıları, MD2, MD4, Haval, Ripe – MD gibi algoritmalardır.

2- Elde edilen mesaj özeti, imzayı atacak kişinin özel anahtarıyla asimetrik olarak şifrelenmektedir. Şifreleme işlemi RSA veya benzeri asimetrik şifreleme algoritmaları kullanılarak yapılmaktadır. Özel anahtarla şifrelenen veriyi ancak anahtar çiftini oluşturan ikinci anahtar olan açık anahtar deşifre edebilmektedir. Özel anahtar sadece imzayı atan kişide saklı tutulur, açık anahtara ise herkesin erişmesi mümkündür.

3- İletinin mesaj özetinin özel anahtarla şifrelenmiş hali iletinin sayısal imza bloğudur ve bu değer iletinin sonuna eklenir, böylece ileti sayısal olarak imzalanmış olur.



Şekil 7. İletinin sayısal imzalanması

1.7.3.3. İletinin İmzasının Doğrulanması

Gelen iletinin sayısal imzasının doğruluğunun kontrolü için, iletiyi imzalayan şahsın açık anahtarına ihtiyaç duyulmaktadır. Açık anahtarlar Sertifika Otoritesi tarafından halka açık yerlerde yayınlanmaktadırlar (Active Directory, LDAP ...).

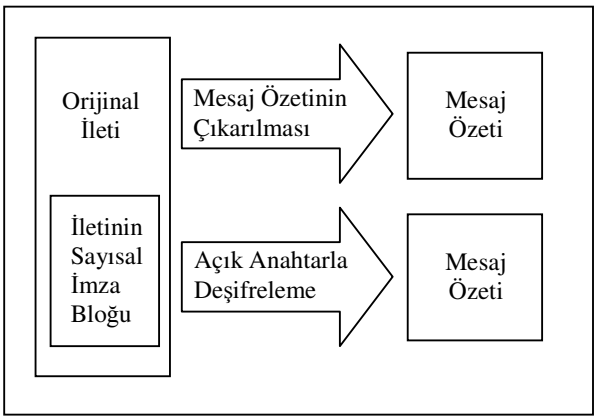
Doğrulama işlemi için gereken adımlar şunlardır :

1- Orijinal ileti, iletinin sayısal imza bloğu haricindeki kısmıdır. Bu kısmın mesaj özeti çıkarılır. Bu işlem için kullanılan algoritma imzalarken kullanılan algoritmayla aynı olmalıdır.

2- İletinin sonuna eklenen imza bloğu imzalayan kişinin açık anahtarıyla deşifre edilir, elde edilen sonuç, orijinal iletinin imzalanması esnasında hesaplanan mesaj özetidir. İletinin deşifre etme işlemi için kullanılan asimetrik algoritma, şifreleme işlemi için kullanılan algoritmayla aynı olmalıdır.

3- 1. ve 2. adımlarda elde edilen değerlerin eşit olması iletinin bozulmamış olmasını ve bu iletiyi imzalayan kişinin de deşifre işleminde kullanılan açık anahtarın sahibi kişi olduğunu göstermektedir.

4- 1. ve 2. adımlarda elde edilen değerlerin farklı olması ise sayısal imzanın geçersiz olduğunu ve iletinin bozulmuş olduğunu göstermektedir [8].



Şekil 8. Sayısal imzanın doğrulanması

1.7.3.4. Sayısal İmzanın İnkar Edilemez Özelliği

Sayısal imzanın inkar edilemez özellikte olmasını, imza doğrulama işleminin 2. adımında sayısal imza bloğunun şifresini çözmek için kullanılan açık anahtar sağlamaktadır. Bu adımda özel anahtarla şifrelenen veri, açık anahtarla deşifre edilmektedir. Sayısal imzalama işleminde kullanılan asimetrik algoritmanın doğası gereği özel anahtarla şifrelenen veri sadece ve sadece anahtar çiftini oluşturan diğer anahtar olan açık anahtarla deşifre edilebilmektedir (aynı şekilde açık anahtarla şifrelenen veri sadece ve sadece özel anahtarla deşifre edilebilmektedir). Eğer bu adımda açık anahtar imza

bloğunu başarıyla deşifre etmişse, imza bloğunu imzalayan kişinin, açık anahtarın sahibi kişi olduğu kanıtlanmış olur. Aksi halde, imza bloğunun açık anahtar tarafından deşifre edilememesi durumunda, imza bloğu bozulmuştur (değiştirilmiştir) veya ileti açık anahtarın sahibi kişi tarafından imzalanmamıştır.

1.8. Kaos ve Kriptoloji

1.8.1. Kaos Nedir?

Kaos teorisi, görünüşte rasgele olan veriler içerisindeki düzeni bulma ile ilgilidir. Kaos, başlangıç koşullarına aşırı duyarlılık gösteren *deterministik* yani, bir sonraki durumun, bir önceki duruma göre belirlendiği sistemlerde, periyodik olmayan davranışlardır. Bir kaotik sistem, rasgele bir sistem değildir. Eğer, sistemin kaotik olduğu belirlenmezse, o zaman o sistem rasgele olur. Örneğin rulet tekerleği kaotik bir sistemdir. Rasgele davranışlar sergilemez. Bir topun yukarıya doğru sıçraması ve sonra tekrar zemine çarpması sonucunda, belirli bir süre sonra bu topun nasıl yükseleceği incelenmek istendiğinde, topun ne kadar yüksekten bırakıldığı, yer çekimi kuvvetinin ne kadar olduğu vb. bilgiler elde edildikten sonra, bu bilgiler bir takım eşitliklere yerleştirilip istenen elde edilebilir. Pratikte, bu tür olaylarda ölçümler yapılmak zorunluluğu olduğundan, yapılan bir ölçüm, gerçeğinden biraz fazla veya az olabilir. Böyle bir durumda da, beklenen sonuç, teorikteki sonuçlardan oldukça farklı olabilmektedir.

1.8.2. Kaosun Ortaya Çıkışı

“Bütünü cebinize koyamazsınız, bunun nedeni cebinizin bütünün bir parçası olmasıdır. Bu nedenle kendi içinde bir boşluğu vardır" [9] .

Bir şeyleri atlamadan irrasyonel bir sayıyı asla yuvarlayamazsınız. Atlamış olduğunuz şey de bilginizdeki bir boşluktur.

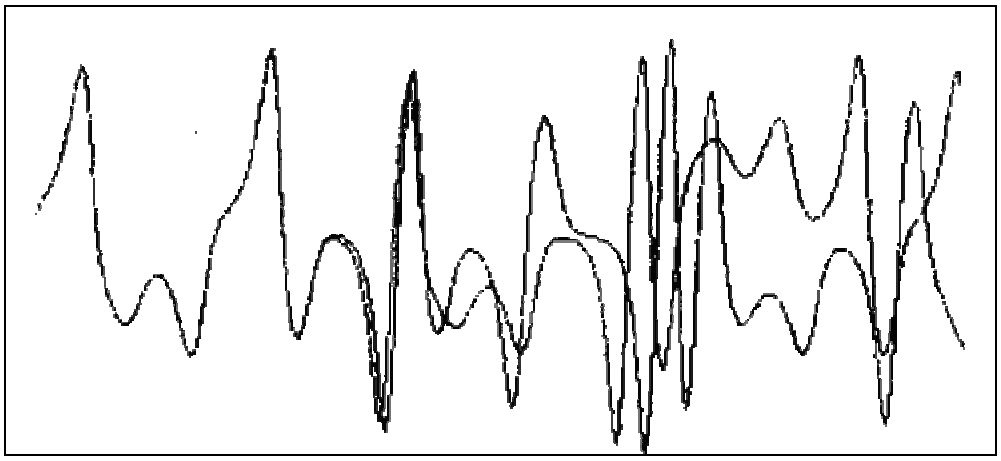
Edward Lorenz ikinci hava durumu tahmininin ilkine uymadığını bulduğunda, sorun bu kayıp bilgiden kaynaklanmaktaydı.

Lorenz ardışık sayı dizilerinin birini uzun uzadıya incelemek istediği bir sırada kestirme bir yol izlemeye kalkışmış, programı tekrar başa dönüp çalıştırmak yerine ortalarda bir yerden başlamıştır. Makineye başlangıç durumundaki şartları vermek için,

daha önce yazıcıdan çıkardığı dizilere bakıp oradaki sayıları klavyeden aynen girdiğinde yepyeni bir bilim dalının ortaya çıkmasına neden olmuştur.

Bilgisayarın yaptığı son dökümde bir önceki dökümün tıpatıp tekrarlanması gerektiği halde, hava durumu bir önceki dökümde yer alan şekilden o kadar hızla uzaklaşmaktaydı ki, bir kaç aylık bir süre zarfında, aradaki bütün benzerlik ortadan kalkmıştı. Sorun şu şekilde açıklanabilir:

Kullanılan bilgisayarın hafızasına kaydedilen ondalık kesir sayıları altı haneliydi: .506127. Yazıcıdan çıkan dökümde ise tasarruf sağlanması amacıyla, sadece üç hane gösterilmekteydi: .506. Lorenz, sayıları kısaltarak son üç rakamı yuvarlamış, binde birlik bir farkın sonucu etkilemeyeceğini düşünmüştü. Şekil 9'da, 0.000127 lik bir farkın belirli bir süre sonra oluşturduğu farklılık gösterilmektedir.



Şekil 9. Lorenz deneyinde başlangıç koşullarına duyarlılık

O dönemlerde bir meteoroloji uydusunun, okyanusun yüzeyindeki sıcaklığı binde birlik bir hassaslıkla okuduğu durum oldukça iyi bir sonuç olarak yorumlanmaktaydı. Lorenz'in Royal McBee'si klasik programı uygulamaktaydı. Tamamıyla determinizme dayalı denklemler sistemi kullanmaktaydı. Belirli bir çıkış noktasından hareket edildiğinde hava durumunun her seferinde tıpatıp aynı gelişmeyi göstermesi gerekirdi. Biraz daha farklı bir çıkış noktasından hareket edildiğinde, hava durumunun da biraz daha farklı bir gelişme göstermesi gerekecekti. Küçücük bir sayı hatasının rüzgârın hafif esintisinden farkı yoktu. Hafif esintiler hava durumunda önemli, büyük ölçekli değişimleri getirmeden önce ya

zayıflayıp ortadan yok oluyorlar ya da birbirlerini dengeliyorlardı. Oysa Lorenz'in kendine özgü denklemler sisteminde küçücük hatalardan büyük felaketler doğmaktaydı.

Karmaşık sistemlerin modellenmesinde bilgisayarların ilk kullanıldığı alan olarak hava durumu tahmini, böylece yepyeni bir alan açılmasını sağlamıştır. Gemi pervanesi tasarlayanları ilgilendiren en küçük ölçekli sıvı akışlarından, ekonomistleri ilgilendiren geniş çaplı para akışlarına kadar her alanda ileriye dönük tahminlerde bulunmayı ümit eden araştırmacılar, ister tabii bilimci olsun, ister sosyal bilimci olsunlar hep aynı tekniklerden yararlanmışlardır. Hakikaten, yetmişli ve seksenli yıllarda, bilgisayarlarla ekonomik durum tahmini yapma konusu global hava durumu tahmini yapmak konusuyla gerçek anlamda bir benzerlik göstermiştir. Modeller, başlangıç durumundaki atmosfer basıncı ya da para arzı gibi şartların ölçümünü gelecekteki trendlerin simülasyonuna dönüştürmek amacıyla düzenlenmiş olan karmaşık denklemlerle içi içe girip karışmıştır. Programcılar, önlenmesi mümkün olmayan sadeleştirme varsayımları, dolayısıyla sonuçların olmayacak şekilde çarptırılması beklentisi içinde olmuştur. Model, Büyük Sahra'yı sel bastırmak ya da faiz oranlarını üç katı yükseltmek gibi göze batacak derecede acayip bir şeyler yapmaya kalkıştığında programcılar denklemleri tekrar elden geçirip, sonucu olması gerektiği şekle sokmuşlardır. Uygulama bakımından, geleceğin getireceği olaylar konusunda ekonometrik modellerin gözleri maalesef bağlı kalmış, üstelik bunu en iyi bilmek durumunda olan pek çok kimse de alınan sonuçlara inanmış gibi davranmışlardır. Ekonomik kalkınma ya da işsizlikle ilgili tahminler açıkça söylenmemekle birlikte iki ya da üç ondalık kesirlik bir hassaslıkla ifade edilmiştir. Hükümetler ve finansal kurumlar, ya ihtiyaç duydukları ya da işsizlikle ilgili tahminleri bedeli karşılandığında sağlamış ve icraatlarını bunlara dayandırmışlardır. Muhtemeldir ki, "tüketicinin iyimserlik katsayısı" gibi değişkenleri "nem oranı" gibi kolaylıkla ölçmenin mümkün olmadığını, üstelik politik hareketler ve moda için henüz mükemmel diferansiyel denklemler yazılmamış olduğunun da farkına varmışlardır. Ne var ki, akış olayını bilgisayarda modelleme süresince çok fazla ümit bağlanamayacağını bilenlerin sayısı pek azdı; hatta mevcut verilerin yeterince güvenilir sayılması ve hava durumu tahmini alanında olduğu gibi, kanunların tamamen fiziksel olması halinde dahi durumda fark olmamıştır.

Bilgisayarlı modelleme evvelce sanat sayılan meteorolojiyi bilime çevirmekte gerçekten başarılı olmuştur. Avrupa'daki merkezce yapılan değerlendirmelere bakılırsa dünyada her yıl hava durumu tahminleri sayesinde milyarlarca dolar tasarruf edilebilir. İstatistik açısından da değerlendirilirse, bu tahminlere sahip olmak elde hiç bir şey

olmamasından daha iyidir. Ne var ki, dünyanın en iyi hava tahminleri bile iki, üç günden öteye gitmemekte, bu süreyi aştığında spekülasyona dönüşmektedir.

Bunun sebebi Kelebek Etkisi'dir. Meteorolojideki küçük olaylar (fırtınalar) açısından bakıldığında her tahmin hızla değer kaybeder. Hatalar ve belirsizlikler çoğalır, zincirleme olaylar halinde gittikçe azalarak anafor ve boralardan sadece uydulardan görülebilecek şekilde bütün kıtaya yayılan burgaçlara kadar şiddetini arttırır [10].

1.8.3. Kaotik Özellikler

1.8.3.1. İterasyon

Çok basit olarak tanımlanmış dinamik sistemlerin davranışları oldukça tahmin edilemez yapıda olabilmektedir.

$$y=x^2 + c \; ; \quad x=y \tag{3}$$

denklemleri ele alınsın. Bu denklemlerden ilki bir parabolü, ikincisi ise, bir doğruyu ifade eder. Bu denklemleri bir düzlemde ifade etmek de mümkündür, bir dizi ifade ile belirtmek de.

1- x verildiğinde, karesi alınır, bu değere bir c sabitinin değeri eklenir ve y sonucu elde edilir.

2- y verildiğinde, direk x elde edilir.

3- 1 adımı, 2 ifadesindeki x değeri ile tekrarlanır.

İlk iki ifade, bir sayının diğer bir sayıya eşitlenmesi olayıdır. Eğer bu işlem reel sayılarla gerçekleştirilirse, reel sayıların reel sayılara map edilmesi sağlanmış olur. 3. adım ise, bu işlemin tekrarlanması sonucunu doğurur. Genel ifade

$$f:x \rightarrow x^2 + c \tag{4}$$

gibidir. x değerinin n kez iterasyona sokulması durumu, $f^n(x)$ ile ifade edilir. Bu iterasyon, bir dizi değerin ortaya çıkmasına neden olur:

$$x, f(x), f^2(x), f^3(x),\ldots,f^n(x)\ldots. \tag{5}$$

Bu diziye yörünge, x'e de kök veya çekirdek (*seed*) denir. x, bu yörüngenin başlangıcıdır. Ne zaman durulacağını söyleyen bir komut olmadığından dolayı, sonsuza kadar değer üretilir. Ta ki, bilgisayar programı sonlandırılana kadar…

c parametresi 0 olduğunda fonksiyon şu hale gelir:

$$f^n \rightarrow \begin{cases} \infty & |x| > 1 \\ 1 & |x| = 1 \\ 0 & |x| < 1 \end{cases} \quad n \rightarrow \infty \qquad (6)$$

x in ± 1 olmadığı durumlarda yörüngeler ya sıfıra ya da sonsuza yakınsar. Sıfır ve sonsuz noktalarına çekici sabit noktalar denir. Çünkü bu noktalar yörüngeleri etrafında toplarlar. ± 1 noktaları da itici sabit noktalardır.

c parametresinin ¼ değerinden büyük olması durumunda, tüm çekirdek değerleri sonsuza ıraksayacaktır. c = ¼ olduğunda ise, parabolle x = y doğrusu ½ noktasında kesişecektir. $|½|$ değerinden büyük çekirdekler sonsuza doğru giderken, ½ <= $|x|$ <= 0 aralığında bulunanlar da, ½ değerine asimptotik olarak yakınsayacaktır. Bu durum, Tablo 9 'da gösterilmektedir.

Tablo 9. $f:x \rightarrow x^2 + ¼$ fonksiyonu için farklı çekirdek değerleriyle oluşan yörüngeler

±1	**±0.75**	**±0.5**	**±0.25**	**±0.1**	**0**
1.25	0.812	0.5	0.3125	0.26	0.25
1.812	0.910	0.5	0.3476562	0.3176	0.3125
3.535	1.078	0.5	0.3708648	0.3508697	0.3476562
12.747	1.412	0.5	0.3875407	0.3731096	0.9708648
162.744	2.246	0.5	0.4001878	0.3892107	0.3875407
26485.994	5.296	0.5	0.4101503	0.4014850	0.4001878
701507907	28.297	0.5	0.4182232	0.4111902	0.4101503
4.921e+17	800.985	0.5	0.4249107	0.4190774	0.4182232
2.421e+35	64158.262	0.5	0.4305491	0.4256258	0.4291070
5.864e+70	4.116e+11	0.5	0.4353725	0.4311573	0.4305491
. . .	. . .	. . .	. . .	. . .	. . .
sonsuz	sonsuz	0.5	0.5	0.5	0.5

Fonksiyonun köklerinden küçük olanı, çekici sabit nokta, büyük olanı ise, itici sabit nokta özelliği gösterir. c = -3/4 için, fonksiyonun kökleri -1/2 ve +3/2 olur. Tablo 10'da,

c=-3/4 değeri için fonksiyonun farklı başlangıç değerleri alınarak iterasyona tabi tutulması sonucunda sergilediği davranışlar gösterilmektedir. 3/2 fonksiyonun köklerinden büyüğüdür ve itici sabit nokta olarak, -1/2 de küçük kök olduğundan çekici sabit nokta olarak davranır. c < -3/4 için, küçük köklere yakınsayan yörüngeler, iki ayrı nokta arasında salınım yapmaya başlar. Bu duruma, çekici sabit noktanın dallanması veya periyot çiftlenmesi adı verilir, artık yörünge daha fazla kararlı davranamaz. Ancak periyodik davranışlara sahiptir. c parametresinin daha da negatif yapılması sonucu, dallanmalar 4, 8, 16 … ∞ şeklinde olur. Ardı ardına gelen dallanmalar arasındaki mesafe sıfıra yakınsar ve bu durumda periyot çiftlenmesi, c<-1/4 şartını sağlayan sonlu bir değerde, sonsuza varır. Bu ana kadar periyodik davranışlar sergileyen yörünge artık periyodik değildir ve tanımlı olduğu aralıkta, her bir noktayı ziyaret eder duruma gelir. Bu davranışa ergodik davranış denir ve kaosun temel karakteristiklerinden biridir [11].

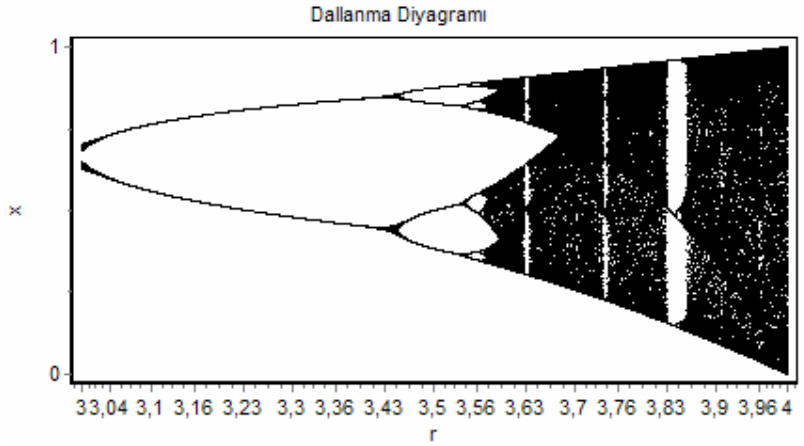
Ek olarak, başlangıçta birbirine oldukça yakın olan çekirdek değerleri, birkaç iterasyon sonrasında, oldukça farklı yörüngeler oluşturmaya başlar. Bu durum, kaosun temel karakteristiği olan, başlangıç koşullarına aşırı duyarlılık olarak adlandırılır. Davranış kaotik davranıştır ve bu durumun olmasına neden olan c değerleri de kaotik bölgeyi oluşturur.

Tablo 10. $f:x \rightarrow x^2$-3/4 fonksiyonunun farklı çekirdek değerleriyle oluşan yörüngeler

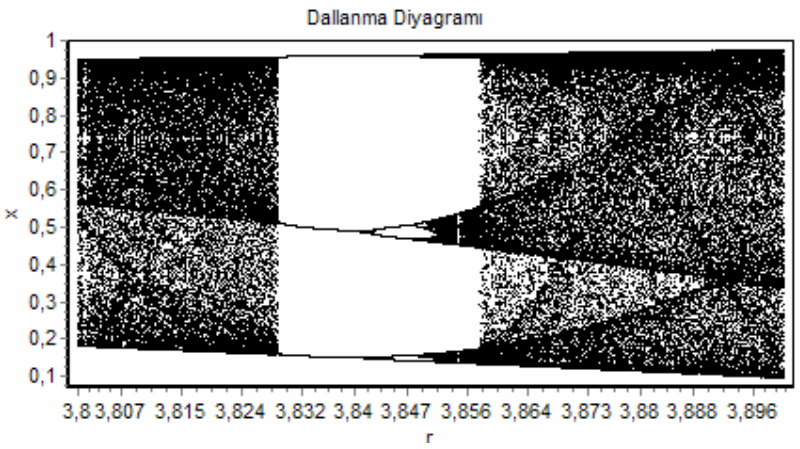
±1.75	**±1.5**	**±1**	**±0.75**	**±0.5**	**±0.25**
2.31	1.5	0.25	-0.1875	-0.5	-0.6875
4.59	1.5	-0.6875	-0.71484375	-0.5	-0.2773437
20.38	1.5	-0.2773437	-0.2389984	-0.5	-0.6730804
414.93	1.5	-0.6730804	-0.6928797	-0.5	-0.2969627
172173.29	1.5	-0.2969627	-0.2699176	-0.5	-0.6618131
2.964e+10	1.5	-0.6618131	-0.6771444	-0.5	-0.3120033
8.787e+20	1.5	-0.3120033	-0.2947537	-0.5	-0.6525639
7.721e+41	1.5	-0.6525639	-0.6650421	-0.5	-0.3240428
5.962e+83	1.5	-0.3240428	-0.3077189	-0.5	-0.6449962
taşma	1.5	-0.6449962	-0.6553090	-0.5	-0.3339798
. . .	. . .	. . .	. . .	. . .	. . .
sonsuz	1.5	-0.5	-0.5	-0.5	-0.5

1.8.3.2. Dallanma Diyagramı

Dallanma kontrol parametreleri değiştirildiğinde, N boyutlu bir çekiciden, 2N boyutlu bir çekiciye olan değişimdir. Periyot çiftlenmesinin görsel ifadesidir.



Şekil 10. Lojistik haritaya ait dallanma diyagramı, r=[3,4]



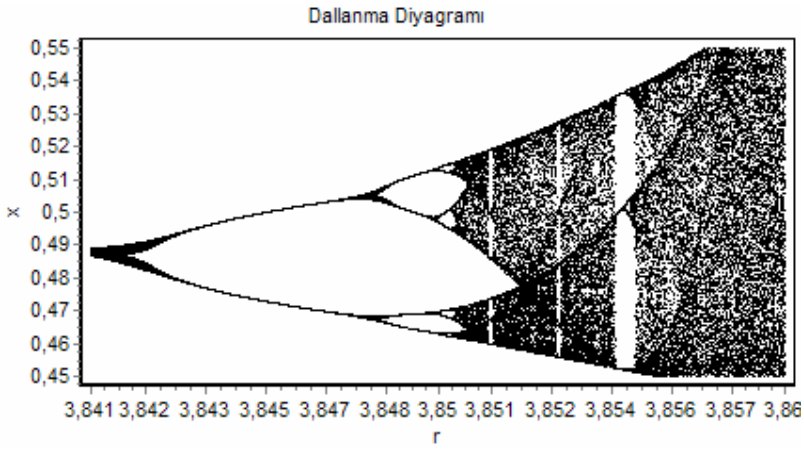
Şekil 11. Lojistik haritaya ait dallanma diyagramı, r=[3.8,3.9]

Dallanma diyagramında, r kontrol parametresinin 1'den küçük değerleri için üretilen değer sıfırdır. r'nin 1 ile 3 arasındaki değerleri için, sistem, tek nokta çekicisi gibi davranır. Ancak, bu çekici nokta, r değerinin artması ile birlikte artış gösterir.

Dallanmalar, r=3, 3.45, 3.54, 3.564, 3.569… noktalarında oluşur. Ancak, sistem, r'nin 3.57 den büyük her değeri için kaotik değildir. Örneğin, r'nin 3.57 den büyük değerleri için diyagram incelenirse, sistemde sadece birkaç x noktasının ziyaret edildiği görülür ve diyagramda beyaz bölgelerin oluşmasına neden olur. r=3.83 noktasında ise, sistemin üç

noktalı bir çekici ürettiği görülür. 3.57 ve 4 noktaları arasında sistem bazen kaotik, bazen de düzenli davranışlar gösterir.

Şekil 10, 11 ve 12'de dallanma diyagramlarından bazı kısımlar gösterilmektedir. Bu şekiller incelenirse, dallanma diyagramlarının belirli bölgeleri alınıp büyütüldüğünde, ana şekle benzerlik gösterdiği görülür. Bu duruma kendine benzerlik denir ve fraktal yapılarda da gözlenir. Kaotik sistemler de fraktal yapıdadır. Ancak her fraktal kaotik özellik göstermez.

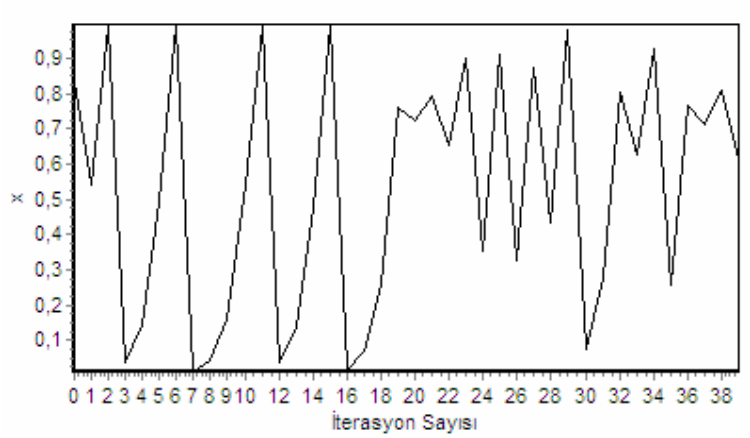


Şekil 12. Lojistik haritaya ait dallanma diyagramı, r=[3.84,3.86]

1.8.3.3. Başlangıç Koşullarına Duyarlılık

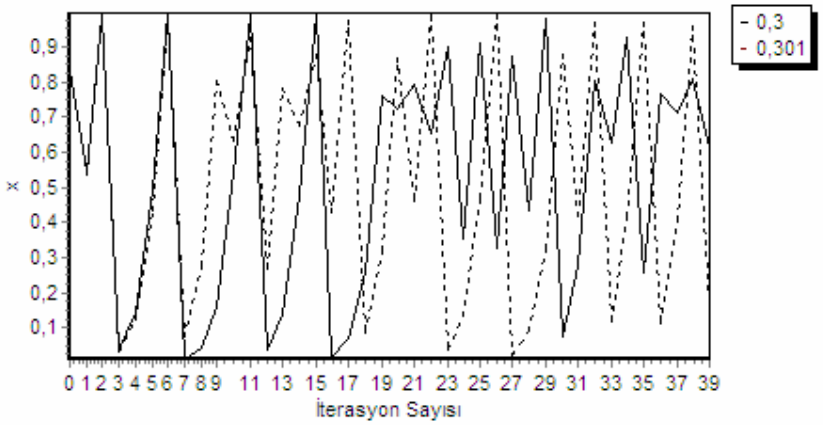
Başlangıç koşullarında yapılan çok küçük bir değişikliğin sonucu aşırı etkilemesi durumuna, başlangıç koşullarına duyarlılık denir. Kaotik ve lineer olmayan sistemlerin tipik özelliğidir. İki benzer değer sisteme başlangıç koşulu olarak uygulandığında, belli bir süre sonra sistemin ürettiği sonuçlar arasındaki fark büyümeye başlayacaktır ve zamanla da bu iki sonuç dizisinin birbirine benzer tarafı kalmayacaktır. Bu olaya kelebek etkisi adı da verilir. Kelebek etkisinde, dünyanın herhangi bir tarafında bulunan bir kelebeğin kanatlarını çarpması, dünyanın başka bir yerinde ve başka bir zamanında bir kasırga oluşmasına neden olmaktadır [12, 13].

Lojistik haritada r=3.99 ve x1=0.3 başlangıç değeri için zaman dizisi Şekil 13'te gösterildiği gibi olmaktadır.



Şekil 13. Lojistik haritada iterasyon sonucu oluşan değerler

Başlangıç değeri olarak 0.301 seçildiğinde sistemin ürettiği değerler, başlangıç koşulu 0.3 olan sisteme göre farklılaşmaktadır. Bu durum, Şekil 14'teki grafikte gösterilmektedir.

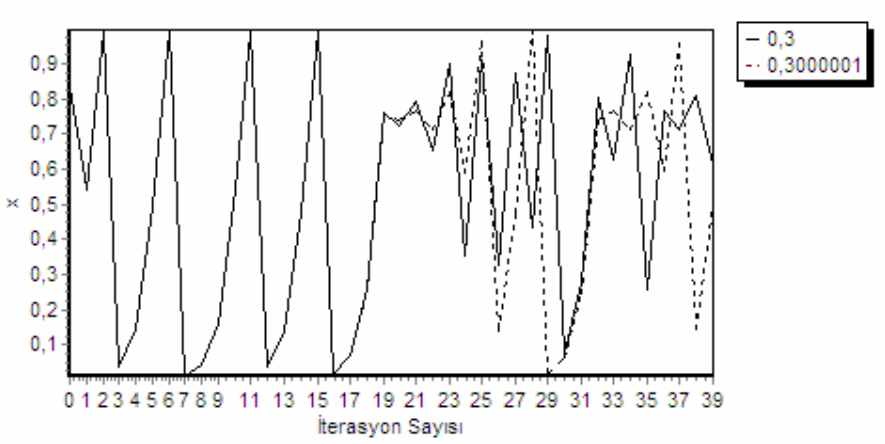


Şekil 14. Başlangıç değerleri arasında 0.001'lik farkın sonucu

Başlangıç değeri 0.3 ile başlangıç değeri 0.3000001 olan iki farklı sistemin ürettiği değerler, yine belirli bir süre sonra farklılaşmaktadır. Ancak bu farklılık, bir önceki örnektekine kıyasla daha geç gerçekleşir. Bu iki sistemin ürettiği değerler, Şekil 15'teki grafikte gösterilmektedir. Sistemler, 24. iterasyondan sonra artık birbiri ile oldukça ilişkisiz değerler üretmeye başlar.

Kaotik bir sistemin davranışlarını gözlemlerken, rasgelelik özelliğinden dolayı bir sonraki durumun tahmin edilmesinin çok zor olduğu ve başlangıç koşullarına göre aşırı

duyarlılığın olduğu önceden bilinir. Çünkü bu özellikler her kaotik sistemin genel özellikleridir ve bunların her biri kendi başına bir sistemin kaotik olduğunu göstermez.



Şekil 15. Başlangıç değerleri arasında 0.000001'lik farkın sonucu

1.8.3.4. Lyapunov Üsteli

x_0 ve $x_0 + \Delta x_0$ uzayda iki nokta olsun. Bu iki nokta, başlangıç koşulu olarak düşünüldüğünde, sistem için iki farklı yörünge oluşturur. Bu yörüngelerden biri referans olarak alınırsa, iki yörünge arasındaki fark, zamanın bir fonksiyonu olur. Ayrıca bu fark, başlangıç noktasının konumuna da bağlıdır ve $\Delta(x_0,t)$ şeklinde ifade edilir. Çekici sabit noktalı veya periyodik noktalı bir sistemde $\Delta(x_0,t)$ değeri, zamana bağlı olarak asimptotik olarak azalır. Eğer sistem kararsızsa, yörüngeler üstel olarak uzaklaşır veya yakınlaşır. Kaotik noktalar için, $\Delta x(x_0,t)$ değeri, kararsız davranır. Başlangıçta, birbirine oldukça yakın olan yörüngelerin zamanla oluşan farklarının ortalama değerini ifade eden değere Lyapunov üsteli denir ve (1)'deki gibi hesaplanır [11].

$$\lambda = \lim_{\substack{t \to \infty \\ \Delta x_0 \to \infty}} \frac{1}{t} \ln \frac{\Delta x(x_0, t)}{|\Delta x_0|} \tag{7}$$

λ, Lyapunov üstelidir ve farklı yörüngeler arasında ayırt edici bir değerdir. Sürekli sistemlerde olduğu gibi, ayrık sistemler için de kullanılan bir sabittir.

λ < 0 : Yörüngeler, kararlı sabit bir noktaya veya kararlı bir peryodik yörüngeye yakınsarlar. Negatif Lyapunov üsteline sahip sistemler, asimptotik kararlılık gösterirler. Daha negatif üsteller, kararlılığı arttırır. Süper kararlı sabit noktalar ve süper kararlı peryodik noktalar, $\lambda = -\infty$ Lyapunov üsteline sahiptir. Bu durum, durmaya yaklaşan salınım yapan bir sarkacın, denge noktasına mümkün olduğunca hızlı bir şekilde gelmesine eş değerdir.

λ = 0 : Yörünge, sabit bir noktadır. Bu durum, sistemin kararlı olduğuna işaret eder.

λ > 0 : Sistem, kararsızdır ve kaotiktir. Çok yakın noktalar, zamanla ayrı konumlara ıraksarlar. Faz uzayındaki tüm noktalar, ziyaret edilir. Ziyaret edilen bu noktalar, kararsızdırlar. Sistem deterministik olmasına karşın, yörüngelerinin davranışlarında herhangi bir düzen yoktur.

Aynı sisteme ait olup, çok az farklı başlangıç koşullarının oluşturduğu zaman serileri arasındaki, E(0) başlangıç farkı, kaotik sistemde zamana bağlı olarak üstel büyüme gösterir. $E(t) = E(0) \exp(L*t)$, bu farkı zamana bağlı olarak gösterir. L, Lyapunov üstelidir, birimi 1/zaman veya frekansdır ve ln(E(t)/E(0)) grafiğinin eğiminden de elde edilebilir. L, kaotik sistemin başlangıç koşullarına göre duyarlılığının derecesini belirtir. Her kaotik sistemin, en az bir pozitif Lyapunov üsteli, çekici sınırlandırıldığından dolayı da, çekiciyi sınırlı tutmak için en az bir negatif Lyapunov üsteli olmalıdır. Fark üstel olarak büyüdüğünden dolayı, başlangıç hatası küçültülerek ileriki durumlar için tahmin yapabilme durumu yine söz konusu olamamaktadır. Örneğin, E(0) değerini küçültmek, sadece ln(100)=4.6 değerinin elde edilme sürecini geciktirir.

1.8.4. Kaotik Sistemlerin Senkronizasyonu

Kaotik sistemler, senkronizasyona meydan okuyan dinamik sistemler olarak yorumlanır. Eğer, iki bağımsız kaotik sistem aynı başlangıç koşulları ile başlarsa, bu şartlardaki en küçük değişiklik, zamanla üstel bir fark oluşturacaktır. Belirli bir süre sonra da iki sistemin hareketleri tamamen ilişkisiz olacaktır.

Pecora ve Carroll'ın 1990'lı yılların başında yaptığı çalışmalar sonucu, iki doğrusal olmayan sistemin, uygun işaret kullanılarak senkron çalışabileceği kanıtlamıştır. Bu çalışmalarda, doğrusal olmayan bir sistem alınmış, bu sistemin bazı alt bölümlerinin kopyası çıkarılmış, kopyalanan ve kopyası alınan alt sistemler, tekrarlanmamış kısımdan alınan uygun bir işaretle sürülerek senkronizasyon olayı gerçekleştirilmiştir. Alt sistem

için, Lyapunov üstellerinin tümü negatif olduğu durumda senkronizasyon sağlanabilmektedir. Senkronizasyondan kasıt şudur: Sistemlerden birinin yörüngeleri, diğerinin aynı değerleri ile birleşmekte ve bu sistemler birbirleriyle bu mesafede kalmaktadırlar. Senkronizasyon yapısal olarak kararlı olmaktadır. n boyutlu dinamik bir sistem ele alınsın.

$$\dot{u} = \mathrm{f}(u) \qquad u = u_1, u_2, \ldots u_n \tag{8}$$

Bu sistem, keyfi iki alt sisteme bölünsün($u = (v, w)$):

$$\dot{v} = g(v, w), \ \dot{w} = h(v, w), \tag{9}$$

Burada;

$$v = (u_1, u_2, \ldots u_m), \ \mathrm{g} = (\mathrm{f}_1(u), \ldots \mathrm{f}_m(u)),$$
$$w = (u_{m+1}, u_{m+2}, \ldots u_n) \text{ ve } \mathrm{h} = (\mathrm{f}_{m+1}(u), \ldots \mathrm{f}_n(u)) \tag{10}$$

şeklindedir. Oluşan iki alt sistemden, w sisteminin kopyası çıkarılıp w' sistemi oluştuğunda sistem şu hale gelmiş olur:

$$\dot{v} = g(v, w), \ \dot{w} = h(v, w), \ \dot{w}' = h(v, w') \tag{11}$$

Artık sistemin boyutu değişmiştir. v sisteminin m, w ve w' sisteminin n-m olan boyutları, sistemin boyutunu 2n-m değerine yükselmiş olur. $v - w$ alt sistemi, w' alt sisteminden bağımsızdır. O nedenle süren sistem olarak kabul edilir ve v işareti de w' alt sistemini sürmek için kullanılır.

Uygun şartlarda, zaman ilerledikçe, $w'(\mathrm{t})$ değişkenleri $w(\mathrm{t})$ değişkenlerine asimptotik olarak yakınsar ve $w(\mathrm{t})$ değişkenleri ile belirli mesafede kalır.

w' nın w''ne yakınsaması için, $t \to \infty$ a giderken, $\Delta w(t) = w'(t) - w(t)$ eşitliğinin sıfıra yakınsaması gerekmektedir. Bu durumun gerçekleşmesi için gerek ve yeter şart, w alt sistemine ait olan alt Lyapunov üstellerinin tümünün negatif olmasıdır [14, 15].

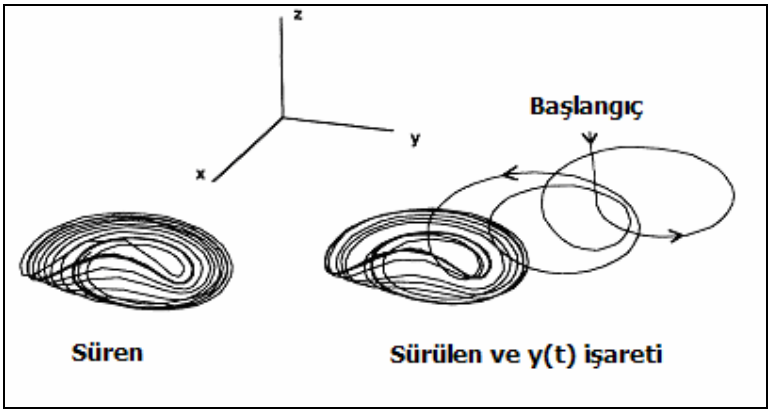
1.8.4.1. Rössler Sisteminde Senkronizasyon

Rössler sistemi, uygun parametreleriyle kaotik davranış sergileyen sistemlerden biridir. (6) sisteminde, (x',z') response rössler sistemini sürmek ve süren sistemin (x,z) bileşenleri ile senkronizasyonu sağlamak için, y bileşenini kullanmak mümkündür.

$$\dot{x} = -y - z$$

$$\dot{y} = x + ay \qquad (12)$$

$$\dot{z} = b + z(x - c)$$



Şekil 16. Rössler süren sistemi için, ($x' - z'$) sürülen sistemi ve y(t) süren işaretinin oluşturduğu çekiciler

Şekil 16'da, kaotik ortamda, bir dizi parametrelerle oluşturulan süren ve sürülen sistemlerin üç boyutlu görüntüleri gösterilmektedir. Görüldüğü üzere, sürülen sistem süren değişkenlerden uzak değerlerde başlangıç değerlerine sahip olmasına karşın, belirli bir süre sonra, süren sistem, çekiciyle senkronizasyonun sağlandığı çekiciye dönüşmektedir.

Rössler sistemi, sadece y işareti ile sürüldüğünde senkronizasyona olanak sağlar. Tablo 11' den de görüldüğü gibi, sadece y süren işareti olduğunda alt Lyapunov üstelleri negatif olmaktadır.

Tablo 11. Rössler sisteminin alt sistemleri, süren işaretleri ve Lyapunov üstelleri

Kontrol Parametreleri	Süren	Sürülen	Alt-Lyapunov Üstelleri
a=0.2, b=0.2, c=9.0	x	(y,z)	(+0.2, -8.89)
	y	(x,z)	(-0.056, -8.81)
	z	(x,y)	(+0.1, +0.1)

1.8.4.2. Lorenz Sisteminde Senkronizasyon

Lorenz sistemi, Edward Lorenz'in meteoroloji tahminlerini gerçekleştirmek amacıyla geliştirdiği sistemdir. Kaotik davranış göstermektedir. 3 boyutludur ve şu şekilde ifade edilmektedir.

$$\begin{aligned} \dot{x} &= \sigma(y - x) \\ \dot{y} &= x(\rho - z) - y \\ \dot{z} &= xy - \beta z \end{aligned} \tag{13}$$

Tablo 12. Lorenz sisteminin alt sistemleri, süren işaretleri ve Lyapunov üstelleri

Kontrol Parametreleri	Süren	Sürülen	Alt-Lyapunov Üstelleri
σ = 10, b=$\frac{8}{3}$, r=60.0	x	(y,z)	(-1.81,-1.86)
	y	(x,z)	(-2.67, -9.99)
	z	(x,y)	(+0.0108, -11.01)

Tablo 12'deki parametreler kullanılarak sistem oluşturulduğunda, x ve y işaretlerinin süren işaret olarak kullanılması durumunda, senkronizasyon sağlanacaktır. Çünkü sadece bu işaretler süren işaret olarak alındığında, sistemin üreteceği alt Lyapunov üstelleri negatif olmaktadır [29].

1.9. Kaos Tabanlı Kriptografi

1990'lı yıllarda yapılan araştırmalar sonucunda, kaos ile kriptografi arasında bir takım benzerlikler olduğu ortaya çıkmıştır. Kaotik sistemlerin bir çok özelliği, kritografi alanında benzer bir özellikle eşleşmektedir. Bu özellikler Tablo 13'te açıklanmaktadır.

Çoğu analog kaos tabanlı kripto sistemler, kaos senkronizasyonu tekniklerine dayalıdır ve gürültülü kanallarda güvenli haberleşmeyi sağlayacak şekilde tasarlanmışlardır. İki kaotik sistem, bir sistemden diğerine gönderilen bir ya da daha fazla süren işaret aracılığıyla birbiriyle senkron çalışabilmektedir.

Kaos, genellikle bir sistemin kendisi ile olan senkronizasyondan çıkması durumudur. Bu durum, periyodik olmayan karmaşık hareketler doğurur. Eğer, iki bağımsız kaotik sistem aynı başlangıç koşulları ile başlarsa, bu şartlardaki en küçük değişiklik, zamanla üstel bir fark oluşturacağı ve belirli bir süre sonra da iki sistemin hareketlerinin tamamen ilişkisiz olacağı daha önce de belirtilmiştir [11, 12].

Tablo 13. Kaos ve kriptografi arasındaki benzerlikler

Kaotik Özellik	**Kriptografik Özellik**	**Açıklama**
Ergodiklik	Konfüzyon	Çıkış, her giriş değeri için benzer dağılım üretir.
Başlangıç koşullarına ve kontrol parametrelerine aşırı duyarlık	Açık metinde veya anahtarda yapılacak küçük değişikle oluşacak farklılıklar. (difüzyon)	Girişteki en küçük değişiklik, çıkış değerlerinde oldukça büyük farklılıklar oluşturur.
Yapısal karmaşıklık	Algoritma karmaşıklığı	Basit bir işlemin karmaşıklık derecesi oldukça yüksektir.
Karıştırma özelliği (*Mixing Property*)	Tüm açık metnin bir kısmında yapılan değişikliğin, yaratacağı etki	Lokal olarak yapılan değişiklikler, uzayın genelinde büyük değişiklikler yaratır.

İki lineer olmayan sistem, kaotik olmalarına rağmen, uygun işaret kullanarak senkronize edilebilir. Pecora ve Caroll'un 1990'lı yıllarda öngördüğü bu yaklaşım, kaotik özellik gösteren bir sistemin, bazı alt parçalarının kopyalarının çıkarılıp, kopyalanmış parçadan alınan işaretle gerçek alt sistemi ve kopyasını sürmekle gerçekleştirilir.

Kaotik şifreleme; kaotik maskeleme, kaotik anahtarlama, kaotik modülasyon ve kaotik kripto sistem olmak çeşitli yollarla gerçekleştirilmektedir [16, 19].

- Kaotik Maskeleme: Analog mesaj işareti, kaotik işaret üretecinin çıkış işaretiyle toplanarak şifrelenir.
- Kaotik Anahtarlama: Sayısal mesaj işareti, birbirinin benzeri iki farklı kaotik çekici kullanılarak şifrelenir. Çekicilerden biri 1 diğeri 0 bilgisini şifrelemede kullanılır. Bu iki çekici, aynı yapıda ancak farklı parametrelere sahip iki kaotik sistem tarafından üretilir. Alıcı tarafta alınan işaret, gönderici tarafta bulunan kaotik sistemlerden herhangi birine benzer olarak oluşturulmuş sistemi sürmek için kullanılır. Mesaj, alçak geçirgen bir fitre kullanılarak deşifrelenir.
- Kaotik Modülasyon: Mesaj işareti, gönderici tarafında bulunan kaotik sistemin bazı parametrelerini modüle etmek için kullanılır. Kaotik sistemin dallanma uzayı çok karmaşık olduğundan, kaotik sistem hakkında bir takım bilgileri olan birisinin, parametrelerdeki bu değişimi bulması çok zordur. Alıcı tarafta, kaotik sistemin parametrelerini elde etmek için bir adaptif kontrol edici kullanılır [28].
- Kaotik Kripto Sistem: Klasik şifreleme teknikleri ve kaotik senkronizasyon kullanılarak data şifreleme gerçekleştirilir. Güvenlik üst seviyededir ve bu şifreleme tekniği henüz kırılamamıştır. Bu yaklaşımda, açık metin, kaotik sistem tarafından üretilen bir işaret kullanılarak belirli şifreleme algoritmalarıyla şifrelenir ve karşı tarafa iletilir. Alıcı tarafta, senkronizasyon bilgisi ile şifreli bilgi ayıklanarak, deşifreleme işlemi gerçekleştirilir [29].

Analog kaotik şifrelemede, mesaj işaretinin diğer tarafa herhangi bir şekilde gönderilmesi durumunda, alıcı, göndericinin kaotik üreteci ile senkronize olmak zorundadır. Bu şekilde, alıcı kaotik işaretini tekrardan üretebilir ve bu sayede de açık metni temsil eden işareti tekrar elde edebilir.

Dijital kaotik şifreleyiciler, dijital bilgisayarlar için geliştirilmişlerdir. Bu sistemlerde, iki ya da daha fazla kaotik harita, sonlu hesaplama duyarlılığı altında, çeşitli yollarla açık metni şifrelemek için kullanılmaktadır. Çeşitli dijital kaotik şifreleyici teknikleri mevcuttur. Bunlardan bazıları şu şekildedir:

- Kaos tabanlı PRNG (sözde rastgele sayı üreteci) tabanlı akış şifreleyiciler.
- Ters sistem yaklaşımlı kaotik akış şifreleyiciler
- İleri ve geri yönde kaotik iterasyon tabanlı blok şifreleyiciler

- Kaotik sözde rastgele sayı dizisinde açık metin bitlerinin aranmasına dayalı kaotik şifreleyiciler [18, 28].

1.9.1. Kaos Tabanlı Kripto Sistemlerin Gerçekleştirilmesi

Her şifreleme türünde olduğu gibi, kaos tabanlı şifrelemede de, sistemin güvenliğini sağlayacak şifreleme hızı ve uygulama maliyeti gibi detaylar önem arz etmektedir. Bu tür detayların başarısızlığı durumunda, güvenlik analizi ve performans değerlendirmesi alanlarında kripto sistemin ne kadar başarılı olduğunun tespiti oldukça güç olmaktadır.

1.9.1.1. Kaotik Kripto Sistemlerin Gerçekleştirilmesi

Kaos tabanlı kripto sistemler, analog veya dijital olarak gerçekleştirilirler. Analog sistemler senkronizasyona dayalı olarak üretilirler ve ilgili kaotik sistemler analog formda gerçekleştirilir. Dijital sistemler ise, senkronizasyondan bağımsız olarak üretilirler ve kaotik sistemler tamamen dijital formda gerçekleştirilir.

Kaotik sistemler kısmen ya da tamamen dijital formda gerçekleştirildiğinde, bir takım dinamik bozulmalar olacaktır. Çünkü dijital kaotik sistemlerin dinamik özellikleri ideal olmayacaktır. En bilinen problem ise, kısa uzunluklu kaotik yörüngelerin sayısının oldukça fazla olmasıdır. Bu da, dijital kaotik şifreleyicinin istenen istatistiksel özelliklerini zayıflatır ve dolayısıyla da güvenliği düşürür. Yapılan çalışmalar sonucunda, bu durumun kaos tabanlı sistemlerde güvenlik açısından oldukça büyük sorun teşkil ettiği görülmüş ve bir takım teknikler ileri sürülmüştür. Bu tekniklerden biri de, kaotik sistemi, küçük bir sözde rastgele işaretle bozmaktır.

Kriptografi alanında iki tür şifreleme yaklaşımı vardır: Birincisi güvenli fakat yavaş bir şifreleyici, ikincisi ise, güvenli fakat hızlı bir şifreleyici üretmektir. Dijital bir kaotik şifreleyici, eğer verimli değilse, kripto analistler tarafından kabul görmeyecektir. Çünkü şifrelemede güvenliğin yanı sıra, performans ve gerçekleştirme maliyeti de oldukça önem taşımaktadır. Gerçekleme ve şifreleyicinin çalışması ile ilişkilendirilen maliyet, hesaplama verimliliği, program boyu ve bellek gereksinimleri de dikkate alınarak belirlenir. (32 bit cpu, 64 bit cpu vs.) . Günümüzde kripto sistemleri değerlendirmek için güvenlik seviyesi, performans ve gerçekleme kolaylığı göz önüne alınmaktadır.

Çoğu, kaos tabanlı güvenli haberleşme sistemlerinde anahtar, sistemin başlangıç şartlarından veya sistem parametrelerinden türetilir. Ancak bu şekilde olsa bile, neyin anahtar olarak kullanılacağı, sınırlarının nasıl olacağı ve duyarlılıklarının ve hassaslığının ne olacağı kesin olarak belirtilmemiştir. Bir anahtar kesin olarak belirlenmelidir.

Bir anahtar uzayı belirlendikten sonra, onu karakterize etmek oldukça önemlidir. O yüzden anahtar uzayı derinden incelenmelidir. Anahtar uzayının boyutu, kripto sistemde erişilebilen ve şifreleme ve deşifreleme için kullanılan anahtarların sayısıdır.

$$K = \{ k_1, k_2, k_3, \ldots k_r \} \tag{14}$$

Burada k_i herhangi bir anahtar, K da, bu anahtarların topluluğudur.

Klasik şifreleme algoritmalarında ki bunlar genelde sayı teorisine dayalı çalışırlar, anahtar, bazı otomatik işlemler sonucu üretilen rastgele sayı dizisinden oluşan bir stringdir. Böyle bir işlemde, eğer anahtar n bit uzunluğunda ise, her n bitlik anahtar, 2^{-n} olasılığında diğerinin bir benzeri olmalıdır. Yani, diğer anahtara çok az benzemelidir.

Çoğu mevcut kaos tabanlı olaylarda, anahtar uzayı doğrusal değildir. Çünkü bütün anahtarlar eşit güçlülükte değildirler. Bir anahtar ile şifrelenen metinlerin deşifrelenmesi, diğer anahtarlarla şifrelenen metinlere oranla daha kolay ise, o anahtar zayıf bir anahtardır.

Bir anahtar olarak birden fazla parametre düşünüldüğünde, bu parametrelerin birbirine bağımlılıkları, hangi aralığın en iyiyi üreteceğini belirlemede zorluklar yaratmaktadır. Kripto sistem tasarlanırken, parametre uzayında hangi bölgelerin kaotik davranışlar gösterdiği belirlenmelidir. Çünkü anahtarlar bu kaotik bölgelerden seçilecektir. Eğer, m parametreye bağlı bir anahtardan söz ediliyorsa, anahtarın seçileceği uzay m boyutlu olmalıdır.

Anahtar uzayını belirlemenin bir yolu da, pozitif Lyapunov üstellerini kullanmaktır. m boyutlu bir dinamik sistem, eğer en büyük Lyapunov üsteli pozitif ise, kaotiktir. En büyük Lyapunov üsteli, seçilen parametrelerin farklı kombinasyonları kullanılarak belirlenebilir. Eğer bu değer pozitif ise, kombinasyon geçerli bir anahtar olarak kullanılabilir.

Düzensiz ve genelde fraktal kaotik bölgeler, çoğu mevcut güvenli haberleşme sistemleri tarafından kullanılmaktadır. Ancak bu durum, kriptografik tasarımlar için yetersizdir çünkü bu bölgelerin sınırını belirlemek kolay değildir. Kriptografi için, oradan seçilecek tüm parametrelerin, sürekli kaotikliği sağlayacağı bir bölge seçilmelidir. Tent haritası, bu bölgeleri karşılayan haritalardan biridir.

$$F(x) = \begin{cases} x/p, & x \in [0, p] \\ (1-x)/(1-p), & x \in [p,1] \end{cases} \quad (15)$$

Burada, $p \in (0,1)$ kontrol parametresidir. Bu aralıktaki her kontrol parametresi için, yukarıdaki parçalı lineer harita, pozitif Lyapunov üsteli verir ve bu nedenle de her zaman kaotiktir. Gerçekte, $F:X \to X$ te tanımlı her parçalı lineer kaotik harita için, eğer her lineer parça, X setine map edilirse, bu map kaotik olacaktır. Bu sonuca dayalı olarak, lineer parçanın özelliklerine bağlı olarak kontrol parametreleri değişmediği sürece, istenen kaotik harita, kaos tabanlı kripto sistemlerde kullanılabilecektir [17, 18].

1.9.1.2. Kaotik Kripto Sistemlerin Avantajları ve Dezavantajları

Kaotik sistem, başlangıç koşullarına duyarlılık gösteren, görünüşte rastgele davranış sergileyen ancak, tamamen deterministik olan bir sistem olarak tanımlanır. Kaosun bu özellikleri, kaotik sistemlerde, uzun süreli tahmin yapmayı zorlaştırmaktadır. Bu durum da, kaosun kriptografi alanında kullanımı için sebep oluşturmaktadır.

Tamamen deterministik olması, aynı fonksiyon ve başlangıç değerlerinin kullanımı durumunda, sürekli olarak aynı sayı değerlerinin üretilmesi anlamına gelir. Rastgele sayı dizisinin tekrardan üretilmesinin söz konusu olmadığı geleneksel rastgele sayı üreteçlerine kıyasla, kaos, aynı fonksiyon ve başlangıç koşullarında aynı değerlerin üretilmesine neden olur. Bu da, sistemin *codebook* gibi sabit anahtarla elde edilen şifreli metinlerin, açık metinlerle eşleştirilmesi ve gelecek farklı şifreli metinlerin anahtardan bağımsız olarak kısmen tahmin edilmesi gibi ataklara karşı dayanıklı olmasını sağlar.

Kaotik sistemlerin başlangıç koşullarına duyarlılık göstermesi sebebiyle, kullanılan kaotik fonksiyonlarda, başlangıç koşullarında yapılacak küçük bir değişiklik, şifreli metinde oldukça büyük değişikliklere neden olacaktır.

Başlangıç koşulları ve parametreler, kullanılan donanıma bağlı olarak, astronomik anahtar uzayı oluşturmaktadır. Virgülden sonra 16 rakama izin veren bir sistemde, sadece bir parametre için 10^{16} farklı durum oluşur. Birden fazla parametrenin kullanıldığı bir sistemde de, durum uzayı oldukça büyük olur. Bu durum da, kaba kuvvet ataklarına karşı sistemi dayanıklı hale getirir.

1.9.1.2.1. Kaotik Kripto Sistemlerin Avantajları

Kaotik fonksiyonların doğası gereği, geleneksel kripto analiz metotları, bu tür şifrelemelerde etkisiz kalmaktadır. Normal kripto analiz metotları, istatistiksel analiz, kaba kuvvet teknikleri vb teknikler kullanarak şifreyi çözmeye çalışmaktadır. Bu tür teknikler, kaotik şifrelemeye uygulanamamaktadır. Çünkü, kaos, rastgele benzeri davranışı dolayısıyla, istatistiksel analize karşı dirençlidir. Anahtar uzayı, oldukça büyüktür. Geleneksel şifreleme sistemlerinde kullanılan anahtarlar, geniş ancak sınırlı integer sayı alanlarından seçilir. Kaotik şifrelemede ise, anahtar olarak, kullanılan donanımın imkanları ölçüsünde, rasyonel sayı aralıklarından seçilen anahtarlar kullanılır. Bu nedenle anahtar uzayı oldukça büyüktür ve kaba kuvvet ataklara karşı dayanıklılık oluşturur.

Kaotik şifrelemenin, geleneksel şifrelemeye göre en büyük üstünlüğü, sayısal-analog dönüştürücü gerektirmeden donanımsal olarak gerçekleştirilmesidir. Çünkü bu dönüşümler ne kadar iyi gerçekleşirse gerçekleşsin, küçük de olsa kayıplara neden olmaktadır. Kaotik fonksiyonun analog olarak gerçekleşmesi için, Van der Pol Duffing isimli bir osilatör geliştirilmiştir. Böylece, mevcut bilgisayar teknolojisi ile sınırlandırılmamış, yüksek hızda çalışan, herhangi bir sorun oluşturmayan bir şifreleme algoritması geliştirmek mümkün hale gelmiştir [20].

1.9.1.2.2. Kaotik Sistemlerin Şifrelemedeki Dezavantajları

Her sistemin bir zayıf noktası vardır ve kaotik şifrelemede kural dışı bir durum yoktur ve belki de, kaotik şifreleme için geleneksel kripto analizlere karşı büyük bir avantaj olan bu durum, dezavantaj haline dönüşebilmektedir. Kripto analizin zor olması nedeniyle, sistemin güvenliği kolaylıkla garanti edilememekte ve güvenliğin seviyesi çok iyi tanımlanamamaktadır.

Analog yaklaşımda, gürültü sorunu ortaya çıkmaktadır. Kaos eşleştirmeleri başlangıç koşullarına duyarlılık gösterdiğinden dolayı, birbirinin aynı ve senkron iki sistemi oluşturmak gerçekten zordur. Çünkü, donanım tabanlı kaos üreteçlerinde, gürültü bu iki sistemin zamanla birbirinden uzaklaşmasına neden olacaktır. Bu durum, iki kaos üretecinin düzenli aralıklarla senkronizasyon yapmasını gerektirmektedir.

Dijital sistemlerin problemleri daha büyüktür. Şifreleme için bir anahtar seçildiğinde veya mesaj bitleri kodlandığında, yapılan şey datanın sınırlı sayı aralığına yerleştirilmesi olayıdır. Bu, sürekli sayı aralığında değil de, sürekli sayı aralığının alt aralığında, sınırlı bir

alanda çalışılması demektir ki, kaotik yörüngelerin periyodik hale dönüşmesine neden olabilmektedir.

Diğer bir sorun da, farklı sayı tanımlamalarının, farklı donanım platformlarında farklı şekillerde ifade edilmesidir. Her ne kadar IEEE'nin belirlediği bir standart varsa da, farklı mikroişlemcilerin farklı veri boyutlarına (4 ila 128 bit) sahip olması, sayılarla ilgili belirli bir işlem standardının oluşturulmasına engel olmaktadır. Böyle bir standart oluşturulması için fonksiyonlar üretilse bile, genel amaçlı mikroişlemcilerde bu fonksiyonları işlemek, fazla işlem gücü gerektirmektedir. Çünkü bu işlemi yapan geleneksel algoritmalar için gerekli fonksiyonların temel işlemleri (aritmetik ve lojik kapılar), zaten mikroişlemcide mevcuttur [20, 26, 27].

2. YAPILAN ÇALIŞMALAR

Bu bölümde, M.S.Baptista türü kaotik şifreleme kullanılarak data şifreleme ve deşifreleme gerçekleştirilmiştir [21, 22]. Şifrelemede, lojistik eşitliğin kaotik özelliklerinden yararlanılmıştır. Lojistik haritanın genel ifadesi şu şekildedir:

$$x_{n+1} = r\, x_n (1 - x_n) \tag{16}$$

2.1. Lyapunov Üsteli Belirleme ve Kaotik Yörünge Üretimi

Alfabetik karakterlerden oluşan bir metnin alıcı tarafına transfer edilmesi için, herhangi bir çekici kullanılabilir. Ancak her çekicinin her bölgesi kaotik davranışlar sergilemez. Bu nedenle, çekicinin kullanılacak bölgesi, mutlaka kaotik davranış sergileyen bölgelerden seçilmelidir. Seçilen herhangi bir bölgenin, kaotik özellik gösteren bölge olup olmadığının anlaşılması için, Lyapunov üsteli hesaplanır. Eğer belirlenen kontrol parametrelerine göre hesaplanan değer pozitif ise, o bölge kaotik özellik gösterir ve şifreleme için kullanılabilir. Aksi halde, sistem ilgili parametrelere göre kararlıdır ve şifreleme için uygun değildir.

Bir sistemin kaotik olup olmadığı, daha ilkel şekilde de belirlenebilir. Basit bir eşitlik alınır, bu eşitlik belirli parametrelerle belirli bir sayıda iterasyona sokulur. Oluşan sonuç serisi yorumlanarak da sistemin kaotik olup olmadığı belirlenir.

Tek boyutlu lojistik haritada r=2 değeri için oluşacak yörünge, periyodiktir. Başlangıç koşulu olarak x_0=0.45 alınırsa,

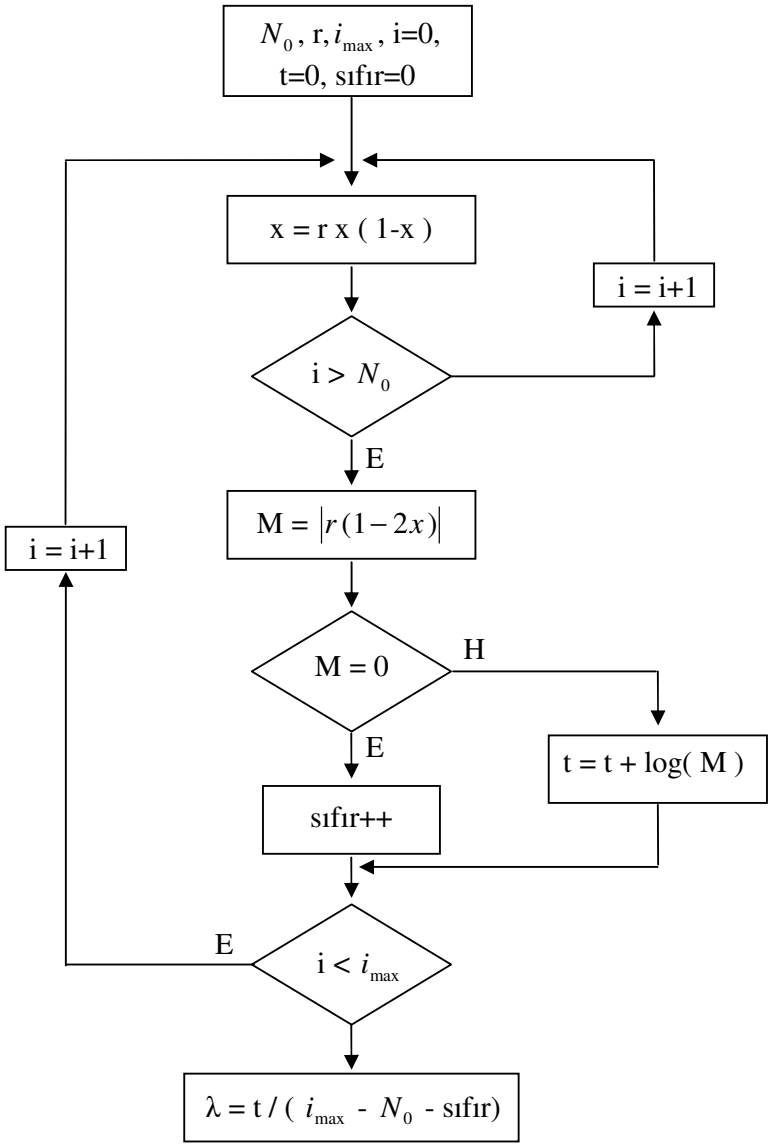
$$\begin{aligned} x_1 &= 2 * 0.45 * (1 - 0.45) = 0.49545 \\ x_2 &= 2 * 0.49545 * (1 - 0.49545) = 0.499959 \\ x_3 &= 2 * 0.\ 499959 * (1- 0.499959) = 0.5 \end{aligned} \tag{17}$$

değerleri elde edilir. Verilen kontrol parametresi ve başlangıç koşuluyla, bu sistem, kararlı duruma sadece 3 iterasyon sonra gelmiştir ve bundan sonraki her iterasyon 0.5 değerini verecektir. Buradan çıkacak sonuç şudur: Sistem, r = 2 için oldukça kararlıdır.

Fakat her sistem, erkenden kararlı duruma geçemeyebilir. Bunun için, binlerce iterasyon gerekebilir ve böyle bir serinin yorumlanması da zordur. Bu nedenle, Lyapunov üsteli kullanılmalıdır.

$$L(c) = \lim_{n\to\infty} \left(\frac{1}{n}\right) \log_2 \left| f'_r{}^n(x) \right| \qquad (18)$$

Burada $f_r(x)$ fonksiyonu, lojistik eşitliği temsil etmektedir. $f^n(x) = f(f^{(n-1)}(x))$'dir. Lojistik eşitliğin türevi, $f'_r{}^n = r - 2rx_n$'dir. Bu eşitliklerden yola çıkılarak, lojistik eşitliğin kontrol parametresine bağlı Lyapunov üsteli Şekil 17'deki gibi hesaplanmıştır [11].



Şekil 17. Lyapunov üsteli hesaplanmasına ilişkin akış diyagramı

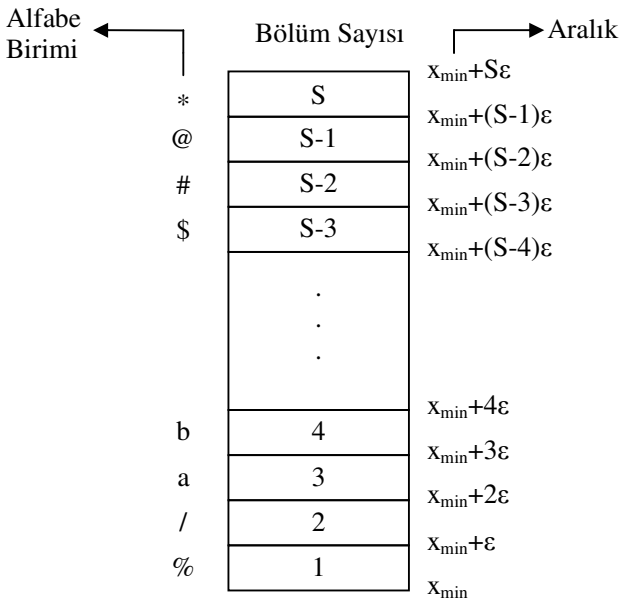
Burada üstelin hesaplanmasında, eşitliğin ilk N_0 iterasyonu, x_0 başlangıç noktasına bağlı olduğundan değerlendirmeye katılmamıştır.

Üstelin hesaplanıp, bölgenin kaotik özelliğe uygunluğu test edildikten sonra, alınan bölge, şifrelenecek metindeki karakterleri temsil edecek şekilde bölümlere ayrılır. Şifreleme, kaotik sistemlerin tipik özelliği olan ergodiklik özelliğinden yararlanılarak gerçekleştirilecektir.

İterasyonlar sonucu üretilecek değerler, [0,1] aralığında olmaktadır ve r kontrol parametresi, sistemi kaotik yapacak şekilde seçilmelidir.

Karakterler şifrelenirken, lojistik harita herhangi bir x_0 değerinden başlatılarak iterasyona tabi tutulur. Bu iterasyonlar, lojistik haritanın üreteceği sonuç, çekicide şifrelenecek karakterle ilişkilendirilen alana varıncaya kadar devam eder. Kaotik sistemin ergodiklik özelliğinden burada yararlanılmaktadır. Daha önce de belirtildiği gibi, ergodiklik, belirli aralıkta bulunan tüm noktalara yalnızca bir kez varılması olayıdır. Fonksiyonu sonsuz kez iterasyona sokun, iterasyon sonucu oluşan değer, fonksiyonun değer ürettiği aralıktaki noktalardan biri olacaktır.

Her harfi temsil eden aralık, fonksiyonun değer ürettiği aralıktır ve iterasyonlar sonucu, bu aralığa mutlaka gelinecektir.



Şekil 18. Çekicinin S adet bölgeye bölünmesi

Şekil 18'de, S adet alfabe karakteri, Sε aralıkları ile ilişkilendirilmiştir. Çekici, S adet bölgeye bölünmüştür. Her bir bölge, $\varepsilon = (x_{max}-x_{min}) / S$ büyüklüğündedir. Her bir bölge de, alfabedeki bir karakter ile ilişkilendirilmiştir. ASCII karakter tablosu kullanılırsa, S=256dır.

Şifreli metni elde etmek için gerekli iterasyon sayısı, Sε aralıkları ile S arasındaki ilişkiye, x_0 başlangıç değerine ve r kontrol parametresine bağlıdır.

x_0 başlangıç noktasıdır ve metinde şifrelenecek her karakter için, her defasında farklı bir sonuç üretilmesi için, yeni bir başlangıç noktası gerekmektedir. i_1, şifrelenecek metindeki ilk karakterin şifreli halinin üretilmesini sağlayan iterasyon sayısı olsun. İkinci karakteri şifrelemek için kullanılacak başlangıç şartı, bir önceki iterasyonda gelinen son noktadır. Aynı şekilde üçüncü karakteri şifrelemek içinse, ikinci karakterle ilgili son iterasyonda üretilen değer başlangıç noktası olarak alınır.

$$\begin{aligned} x_1 &= f^{i_1}(x_0) \\ x_2 &= f^{i_2}(x_1) \\ x_3 &= f^{i_3}(x_2) \end{aligned} \tag{19}$$

Bu durum genellenirse, aşağıdaki ifade ortaya çıkar.

$$x_n = f^{i_n}(x_{n-1}) \tag{20}$$

Başlangıç koşullarının durumu, şifrelenecek metindeki farklı karakterler için, aynı şifreli metinlerin üretilmesini hedefler. Bu durum, kriptografide bire bir olmayan şifreleme olarak adlandırılmaktadır.

$$x_2 = f^{i_1+i_2}(x_0) \tag{21}$$

x_2 (21)'deki gibi de ifade edilebilir. Ancak, başlangıç koşulunun bu şekilde belirlenmesi durumunda iterasyon sayısı zamanla 65536 değerini aşacaktır. Bu da performans açısından verimsizlik yaratır. Şifrelenen karakter, x_0 başlangıç ve r kontrol parametresinin yanı sıra, N_0 geçiş zamanına bağlıdır. Geçiş zamanı, Lyapunov üstelinin

hesaplanmasında da dikkate alınan ve sistemin başlangıç noktasının yaratacağı bağımlılığı en aza indirmek için kullanılan bir parametredir.

Bir karakteri şifrelemek için kullanılacak iterasyonların sayısı farklılıklar göstermektedir. Yani, bir karakter sadece belirli bir uzunluktaki yörünge kullanılarak şifrelenmez. Çünkü herhangi bir x_0 başlangıç değerinden başlanarak hedef ε aralığına, farklı boyutlu yörüngelerle varılabilir. Sonuç olarak da, şifrelenecek metindeki bir karakter, farklı şekillerde şifrelenebilir. Ancak, kullanılacak yörüngenin boyunun büyük olması, uygulama açısından hızı azaltıp, oluşacak dosya boyutunu artıracağından dolayı verimsizlik yaratır.

Düşük boyutlu yörüngelerin tercih edilme sebebi, kaotik çekicilerin sabit yoğunluk fonksiyonudur. Bu sabit yoğunluk, sonsuz boyutlu yörüngelerin uzaydaki dağılımıdır. Kaotik sistemler ergodik olduğundan, herhangi bir başlangıç değeri ile iterasyona tutulan bir sistem, çekicideki her ε aralığına birçok kez varacaktır ve bu aralıkların ziyaret edilme oranı, bu sabit yoğunluk oranına bağlı olarak değişecektir.

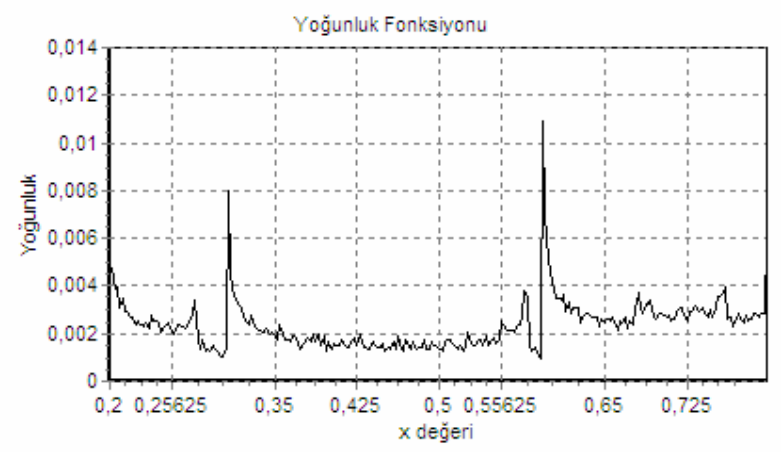
Sabit yoğunluğu hesaplamak için, x_0 başlangıç noktasından başlayıp N kez iterasyona tabi tutulan fonksiyonun, çekicinin bölündüğü ε aralıklarına kaç kez uğradığı belirlenir. Bu sayılar da N değerine bölünerek, her bir aralık için yoğunluk değeri belirlenmiş olur.

$r=3.78$, $x_n=0.432031250$ alınan bir lojistik haritanın sabit yoğunluğu Şekil 19'da gösterilmektedir. Burada lojistik haritanın [0.2,0.8] aralığı, her biri 0.002343750 uzunluğunda olan, 256 adet ε aralığına bölünmüştür. Eşitlik, x_0 başlangıç noktasından itibaren 65536 kez iterasyona tabi tutulduğunda, oluşan en düşük yoğunluk değeri olan 0.0011 değerine sahip nokta sayısı 76 olarak bulunmuştur.

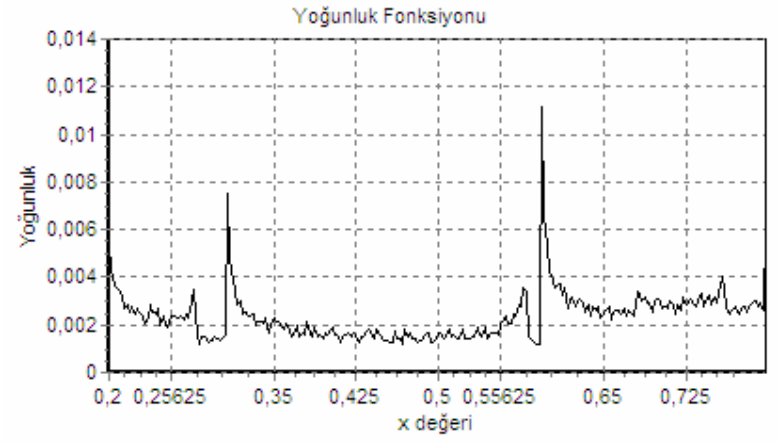
Şekil 19 ve Şekil 20 incelendiğinde, başlangıç noktaları farklı olan yörüngelerin, lojistik eşitliğin [0.2,0.8] aralığında oluşturacağı yoğunluğun, benzer olduğu görülmektedir. Bunun anlamı şudur: Farklı başlangıç koşulları, benzer sabit yoğunluk değerine sahip olurlar ve 65536 iterasyonlu yörünge, hangi başlangıç noktasından başlanılırsa başlansın, $r=3.78$ değeri ile, her ε aralığına en az 76 kez uğrar.

Yörüngenin boyutu, 65536 değeri ile sınırlandırılmıştır. Bunun yanında geçiş zamanından küçük olan iterasyonlar da ihmal edilmiştir. Bundan dolayı, $b=3.78$ için her ε aralığını en az 76 kez ziyaret eden bir yörüngeye rastlanılamayabilinir. Eğer istenmeyen bir kişi tüm S anahtarlarını ve bu anahtarlarla ilişkilendirilen S_ε aralıklarını bilirse, x_0 başlangıç değeri ve r kontrol parametresini bilmediğinden dolayı, şifreyi kıramayacaktır. Çünkü, aynı S değerleri kullanılarak, r ve x_0 değerlerinden 10^{-16} kadar küçük bir farkla

başlatılan bir işlem, birkaç yüz iterasyondan sonra bir diğerine göre oldukça farklı yörüngeler oluşmasına neden olacaktır. Bu, kaotik sistemlerin başlangıç koşullarına olan aşırı duyarlılığının bir sonucudur. Virgülden sonra on altı rakam duyarlılığına sahip bir uygulamada, bilinmeyen x_0 ve r kontrol parametrelerinin deneme yanılma yöntemiyle çözülmesi durumunda durum uzayı, 10^{16} x 10^{16} büyüklüğündedir.



Şekil 19. Lojistik haritanın, r=3.78, N=65536 x_n= 0.432031250 şartlarıyla oluşan sabit yoğunluk değerleri



Şekil 20. Lojistik haritanın, r=3.78, N=65536, x_n= 0.648565759 şartlarıyla oluşan sabit yoğunluk değerleri

Şifrelenmiş karakter, geçiş zamanı ile 65536 arasında sınırlandırılmış olsun. Yoğunluk dağılımından da anlaşıldığı gibi herhangi bir karakter, farklı şifreli karakterlerle

eşleştirilebilir. r=3.78, N=65536 x_n= 0.432031250 şartlarıyla elde edilen yoğunluk fonksiyonunda, minimum yoğunluk değerine sahip nokta sayısı 76 idi. Bu şartlar kullanılarak bir şifreleme yapıldığında, herhangi bir karakteri şifrelemek için minimum 76 farklı yörünge ortaya çıkar. Her yörünge, farklı bir şifreyi temsil ettiğinden dolayı da, gönderici, bu seçenekler arasında tercih yapmak zorundadır.

2.2. Yörünge Seçimi

Yörünge tercihi yapılırken, en küçük uzunluklu yörünge de seçilebilir, diğer yörüngelerden herhangi biri de. En küçük uzunluklu yörünge, N_0 geçiş sayısından büyük olan ve iterasyonlar sonucu, karakterle ilişkilendirilen aralığa ilk kez varan yörüngedir. Eğer, şifreli karakterin belirlenmesinde en küçük yörünge kullanılmak istenmiyorsa, iterasyonlar devam ettirilir. Karakterle ilişkilendirilen aralığa en az minimum yoğunluk değerine sahip nokta sayısı kadar varılacaktır. İterasyonlar devam ettirilirken, belirli bir fonksiyon kullanılarak yörüngelerden herhangi biri seçilerek işlem durdurulur ve karakterin şifresi olarak, o ana kadar yapılan iterasyon sayısı kabul edilir.

2.3. Şifreleme

Şifreleme işlemi, lojistik harita kullanılarak, geçiş zamanı olarak N_0 = 256, başlangıç koşulu olarak x_0 = 0.306, kontrol parametresi olarak r=3.7801 ve çekicinin kullanılacağı aralık olarak da [0.2,0.8] alınarak, "güvenlik" kelimesinin üzerinde örnekle açıklanacaktır.

İlk olarak, çekicinin kullanılan aralığı 256 eşit parçaya bölünür ve her bir aralıkla bir harf eşleştirilir. Eşleştirme sonucu, x. aralık, ASCII tablosundaki karakterlerden, sayısal değeri x olan karakteri temsil etmektedir.

Yoğunluk fonksiyonu yorumlandığında, minimum yoğunluk değeri yaklaşık olarak 0.00102234 olmaktadır ve bu minimum değerin oluşmasına neden olan 67 adet nokta vardır. Bunun anlamı şudur: [0.2, 0.8] aralığındaki her alt aralığa en kötü ihtimalle 67 kez gidilecektir. Diğer bir deyişle, bu aralıklardan birine gidilmeme olasılığı 0 dır.

Kelimedeki karakterlerin ASCII değerleri, çekicide bu değerlere karşılık gelen aralıklar Tablo 14'te gösterilmektedir.

Tablo 14. Şifrelenecek karakterlerin, lojistik harita uzayında eşleştirildikleri aralıklar

Harfler ve ASCII Değerleri		Alt Aralık	Üst Aralık
g	103	0.44140625	0.44375
ü	252	0.790625	0.79296875
v	118	0.4765625	0.47890625
e	101	0.43671875	0.4390625
n	110	0.4578125	0.46015625
l	108	0.453125	0.45546875
i	105	0.44609375	0.4484375
k	107	0.45078125	0.453125

Şifreleme işleminde, şifre olarak, karakterler için yapılan iterasyonlarda sonuca gidilen ilk değer yani geçiş zamanından büyük, en küçük uzunluklu yörünge alınır.

İlk olarak fonksiyon geçiş zamanı kadar iterasyona sokulur. x_{257} başlangıç değerinden başlanarak, ilk harf olan g harfi ile ilişkilendirilen aralığa varıncaya kadar, (16)'daki iterasyon gerçekleştirilir. $x_0 =$ 0.306 başlangıç noktasından, g harfinin temsil edildiği [0.44140625, 0.44375] aralığa gelinceye kadar, 861 iterasyon gerekli olmaktadır. 861, g harfinin şifreli halidir ve dosyaya iki bayt şeklinde yazılır. Bir sonraki harfin şifrelenme işleminde ise, başlangıç noktası değişmiştir ve 0.442281327377344 değerini almıştır. Bu değer, 861. iterasyon sonucunda fonksiyonun ürettiği değerdir. Bu noktadan başlanarak fonksiyon 308 kez iterasyona sokulduğunda, ü harfini temsil eden [0.790625, 0.79296875] aralığından bir değer üretilmiş olunur. Tüm bu işlemler sonucu oluşan değerler, Tablo 15'te gösterilmektedir.

Tablo 16'da ise, en küçük yörünge uzunluğu şifre olarak alınmayıp, rastgele olarak yörünge uzunluğunun belirlenmesinde oluşan değerler gösterilmektedir. Bu yaklaşımın en büyük avantajı, uzun bir şifreli metnin frekans dağılımını düzleştirmesidir. Bu düzleştirme etkisi çok önemlidir. Çünkü, bu metot kullanılarak oluşturulan şifreli metnin dağılımı, ne metnin diline(Türkçe İngilizce veya ratsgele seçilmiş bir metin), ne de başlangıç koşuluna bağlıdır. Dağılımı etkileyen şey, r kontrol parametresidir. Bu yüzden, frekans dağılım analizi yaparak kontrol parametresinin yaklaşık değerini bulmaya çalışan bir kişi, bu nedenle başarısız olacaktır.

Tablo 15. En küçük yörünge seçilmesi durumunda alınan sonuçlar

Harfler ve ASCII Değerleri		Başlangıç Değeri	İterasyon Sayısı	Gelinen Son Nokta
g	103	0.306	861	0.442281327377344
ü	252	0.442281327377344	308	0.792454151968555
v	118	0.792454151968555	3476	0.478455215303413
e	101	0.478455215303413	377	0.436956518098058
n	110	0.436956518098058	382	0.459665757003935
l	108	0.459665757003935	391	0.453199321772555
i	105	0.453199321772555	689	0.447855666552281
k	107	0.447855666552281	432	0.452970902511186

Tablo 16. Yörüngelerden herhangi biri seçilmesi durumunda alınan sonuçlar

Harfler ve ASCII Değerleri		Başlangıç Değeri	İterasyon Sayısı	Gelinen Son Nokta
g	103	0.306	4175	0,442637808284061
ü	252	0,442637808284061	822	0,792828280870799
v	118	0,792828280870799	803	0,477177832469876
e	101	0,477177832469876	3345	0,437865863981685
n	110	0,437865863981685	1648	0,458814830115611
l	108	0,458814830115611	1768	0,453204057005919
i	105	0,453204057005919	1633	0,447457507249052
k	107	0,447457507249052	750	0,452072386575221

Şifre olarak her zaman en küçük yörünge uzunluğu alınacak olursa, kontrol parametresini ve başlangıç şartını değiştirmeden yapılacak her şifreleme aynı sonucu üretecektir. En küçük yörüngenin seçilmemesi durumunda, kontrol parametresi ve başlangıç şartı değiştirilmeden 67^5 farklı şifre üretilebilir. Başlangıç koşulunun değiştirilmesi durumunda ise, 10^{16} x 67^5 adet farklı şifre elde edilir. Her ikisi değiştirildiğinde ise olası farklı şifre sayısı, 10^{16} x 10^{16} x 67^5 olarak değişmektedir.

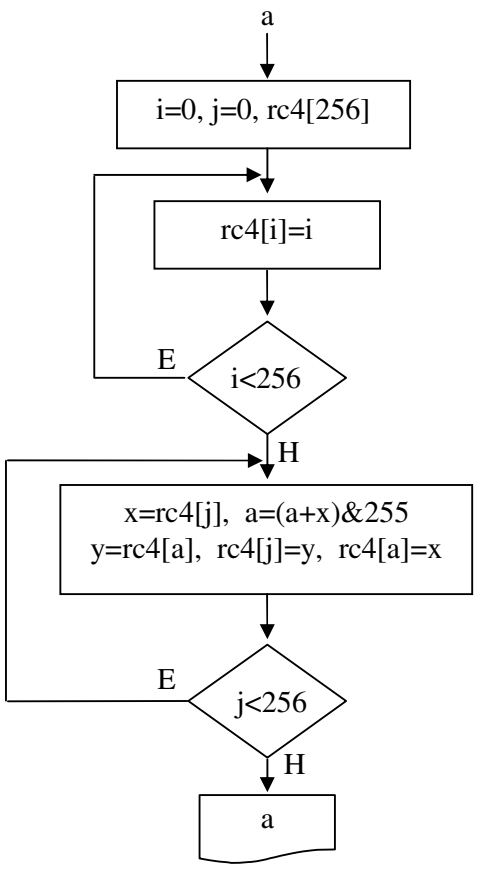
Bu şifreleme işlemi sonucu oluşan karakterler düzgün dağılıma sahip değildir. Düzgün dağılım, farklı karakterlere ait şifrelerin meydana gelme olasılığının eşit olması durumudur. Şifreleme sonucu ortaya çıkan şifrelerin gösterdikleri dağılım düzgün olmadığında, şifreli metin hakkında daha kolay yorum yapılabilir. Bu tür şifrelemeler, ataklara açıktır.

Kullanılan algoritmada olası şifreler, geçiş zamanından büyük değerler alır. Şifre için başlangıçta 65535 üst sınırı düşünülmüşse de, iterasyonların bu değerde sonlanmama ihtimalinin mevcut olması sebebiyle bu sınırlamadan vazgeçilmiştir.

Oluş olasılığı, ataklar için ön koşuldur. O nedenle, düzgün olmayan şifre dağılımı, oluşan şifre karakterlerinin herhangi bir sözde rasgele sayı (*pseudo random number*) dizisiyle maskelenerek gizlenebilir.

Sözde rastgele sayılar, sözde rasgele sayı üreteçleri (*Pseudo Random Number Generator*) tarafından üretilirler. Bu sayılar gerçekte rasgele değildir. Bu sayılar, rastgele olarak oluşturulan sayıların bir takım özelliklerini içerirler. Gerçek rastgele sayılar, donanımlar aracılığıyla üretilirken, sözde rasgele sayılar matematiksel ifadelerle üretilirler ve data şifrelemede oldukça önemli yere sahiptirler.

Şekil 21'de, sözde rasgele sayı üretmede kullanılan algoritmalardan biri olan RC4 algoritmasının akış diyagramı verilmiştir [23].

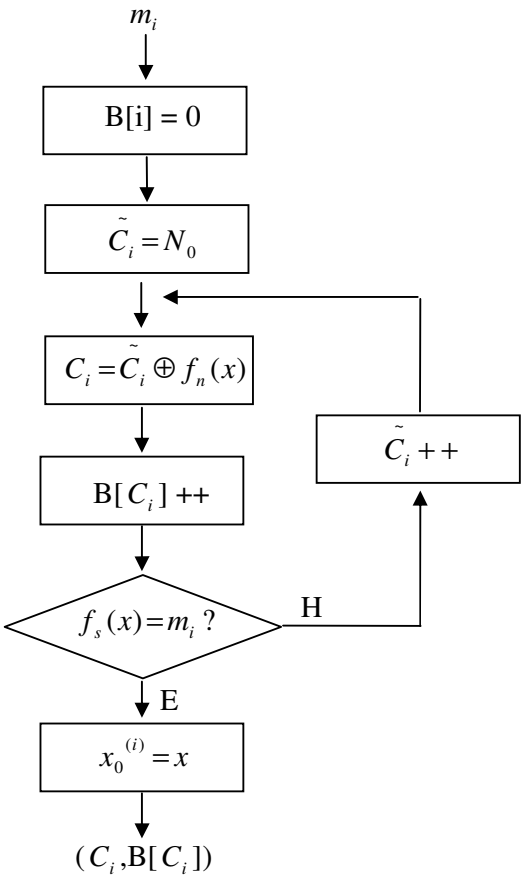


Şekil 21. RC4 algoritmasının akış diyagramı

Sözde rasgele sayı üreteci, f(x)=y gibi bir fonksiyon da olabilirdi. Ancak, bu fonksiyonların ürettiği sayı dizileri, şifre karakterlerinin maskelenmesi amacıyla kullanılacağı için, bu kadar basit fonksiyonlar, kriptoloji açısından verimsizlik yaratır. O nedenle, uygulamada RC4 algoritması kullanılmıştır [23].

Bu maskeleme sonucu, şifreli karakterler gizlenmiş olur. Ancak, bu sefer de farklı bir hatayla karşılaşılır. Maskeleme işlemi, i. karakterin şifrelenmesi sonucu elde edilen C_i karakterinin uygun bir $f_n(x)$ fonksiyonu ile XOR işlemine tabi tutulması ile gerçekleştirilir. Burada ortaya çıkan problem şu şekilde açıklanabilir:

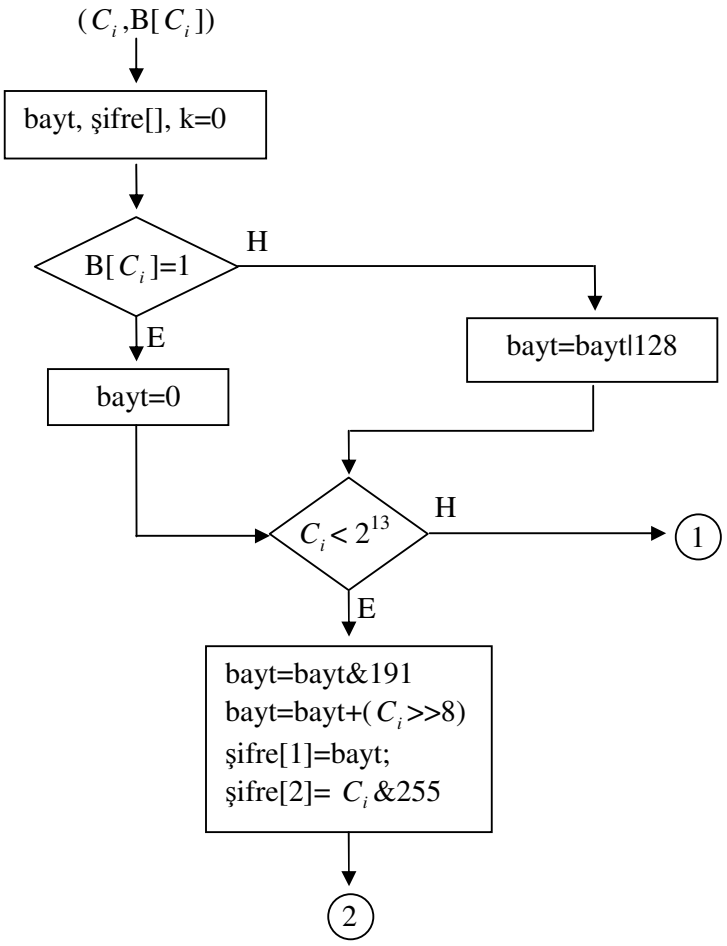
$C_i \oplus f_n(x)$ nini oluşturacağı değer, çok küçük bir olasılıkla da olsa, $\tilde{C}_i' \oplus f_n(x')$ ifadesi tarafından da oluşturulabilir. Bu problem, şifreleme boyunca her iterasyon için $C_i \oplus f_n(x)$ değeriyle kaç kez karşılaşıldığının kaydedilmesi ile aşılabilmektedir (B[C_i]). Şekil 20'de şifreleme için kullanılan algoritma gösterilmektedir.



Şekil 22. Şifreleme Algoritması

Oluşan şifre karakterleri, geçiş zamanından büyük değerler almaktadır. Geçiş zamanı için ortalama bir değer belirlenir ve oluşan şifreler, tek baytla ifade edilecek kadar küçük değildirler. O nedenle, karakterin şifresi, dosyada minimum iki bayt alan kaplamaktadır. Bu durum da, şifreli metin boyutunun orijinal metnin iki katına çıkması problemini ortaya çıkarmaktadır. Bir diğer problem de, B[C_i] değerinin dosyada saklanması durumunda ortaya çıkmaktadır. B[C_i] değeri, çok küçük bir olasılıkla 1 den farklı değer almaktadır. O nedenle, değeri bu kadar küçük olan bir sayı için, dosyaya bir baytlık bilgi yazmak, verimlilik açısından olumsuzluk yaratacaktır.

C_i şifre değeri dosyaya olduğu gibi yazılmaktadır. Ancak, B[C_i] için, gerektiği durumlarda yazılmasını sağlayacak bir yöntem kullanılmıştır. Bu yöntemde, C_i şifre değerinin en anlamlı iki biti de kontrol biti olacak şekilde kullanılmıştır.

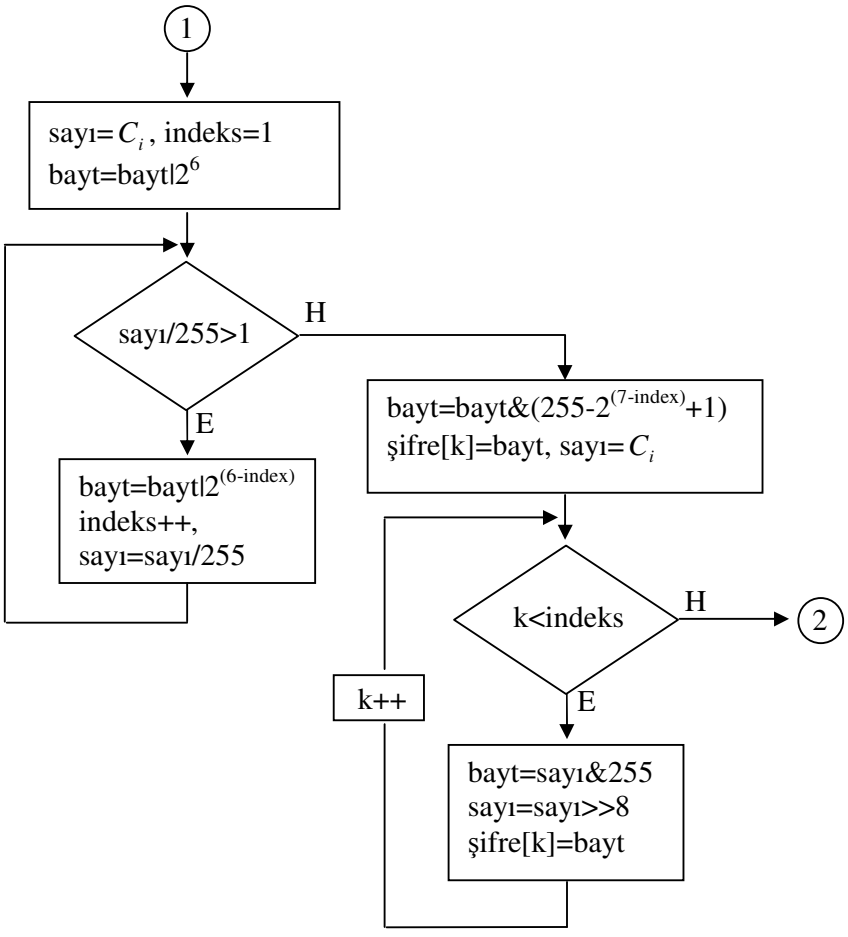


Şekil 23. Sınır değerlerinin aşılmaması durumu

Algoritmanın birinci bölümü iterasyon sayısının belirlenen değeri aşmaması durumunda, dosyaya hangi değerlerin yazılacağını göstermektedir. Burada sınır değeri, 2^{13} olarak belirlenmiştir. Bunun sebebi, baytın geri kalan iki bitinin kontrol biti olarak kullanılmasıdır. Bu durum herhangi bir olumsuzluk yaratmaz. Çünkü, sınır değerinin aşılması nadiren gerçekleşen bir olaydır. Bu durumun kontrolünü yapan bitler için fazladan bit kullanımı bu şekilde önlenmiş olur.

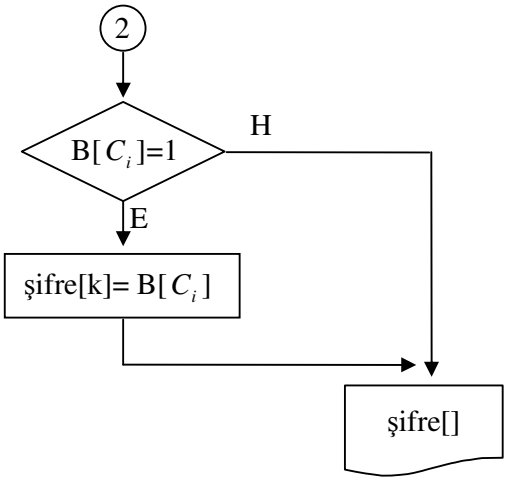
Belirlenen sınır değerinin aşılması durumunda ise, farklı bir yöntem kullanılmaktadır:

Sınır değeri aşıldığında, dosyaya yazılan ilk bayt, tamamen kontrol amaçlı kullanılmaktadır. Şöyle ki, bu baytın en anlamlı biti, b kontrol parametresi için kullanılmaktadır. En anlamlı bitten en anlamsız bite doğru olan tüm 1 değerli bitler, şifre değeri için, dosyaya kaç adet bayt yazıldığını ifade eder. Bu işlemin akış diyagramı Şekil 24'te gösterilmektedir.



Şekil 24. Sınır değerlerinin aşılması durumu

Üçüncü ve son bölümde ise, b kontrol parametresinin değerine göre dosyaya bir bayt daha yazılıp yazılmayacağı belirlenir.



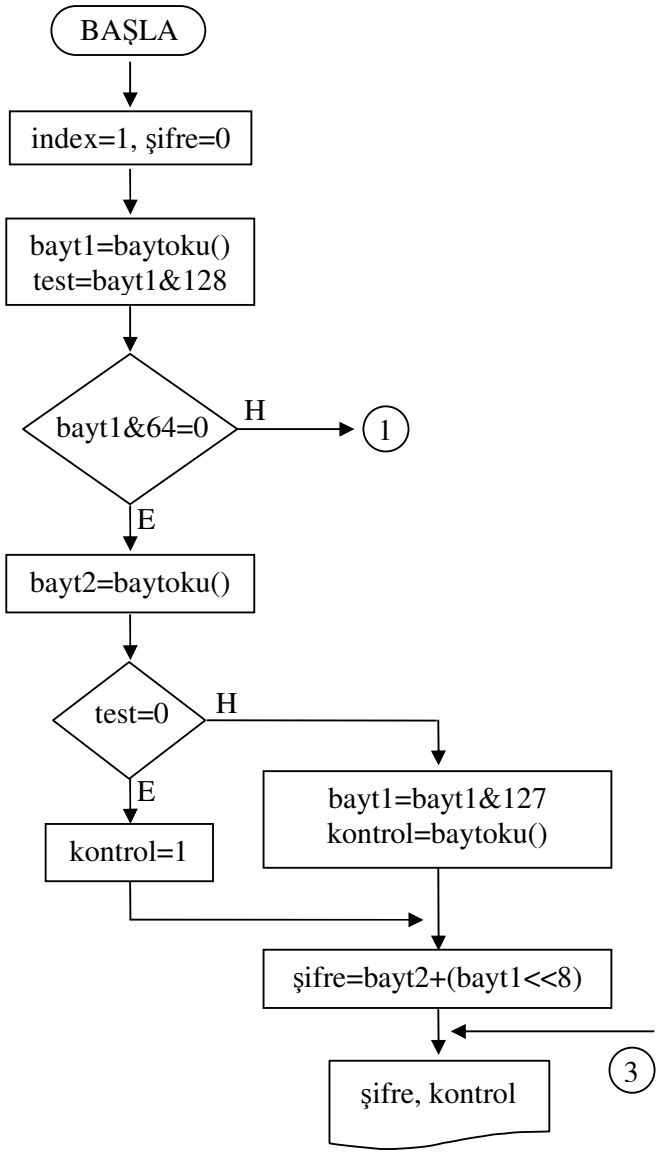
Şekil 25. B kontrol değerinin dosyaya yazılması

2.4. Deşifreleme

Şifreleme sonucu oluşturulan dosyada, deşifreleme için gerekli bilgiler vardır. Deşifreleme işlemi, şifrelemenin tersi gibidir. Şifrelemede yapılan işlemler, deşifrelemede, tersten başlanarak gerçekleştirilir. Şifrelemede olduğu gibi, bu işlemde de, her bir harf, çekicinin, şifrelemede de kullanılan belirli bir aralığıyla ilişkilendirilir. Şifrelemede kullanılan kontrol parametreleri ve başlangıç koşulları aynen alınır. Dosyadan gerekli bilgilerin okunması sonrasında, sadece elde edilen değer kadar iterasyon gerçekleştirilerek, iterasyonlar sonrasında, karakter tablosunda hangi aralığa gelindiği bulunur ve bu şekilde deşifreleme işlemi gerçekleştirilir.

Dosyadan bilgiler bayt olarak okunur ve yorumlanır. Şifrelenmiş her bir karakter, dosyada en az iki baytla temsil edilmektedir. Bu durum, iterasyonların boyunun sınırlandırılamamasının bir sonucudur. Dosyadan okuma işleminde izlenen yöntemin birinci kısmı, Şekil 26'da algoritmik olarak gösterilmiştir.

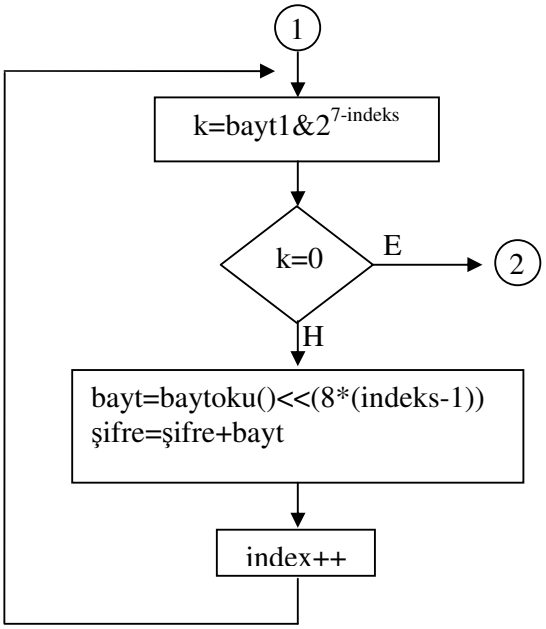
Bu kısımda, şifreleme esnasında belirlenen maksimum değerin aşılmadığı durumda, dosya okumanın ne şekilde gerçekleştiği anlatılmaktadır. Maksimum değerin aşılmaması, dosyadan iki bayt okunacağı anlamına gelmektedir. Bu durum, dosyadan okunan ilk baytın en anlamlı ikinci bitinin kontrolü ile test edilir. En anlamlı ilk bit ise, B kontrol değerinin dosyada mevcut olup olmadığının testinde kullanılır.



Şekil 26. Taşma olmaması durumunda okuma

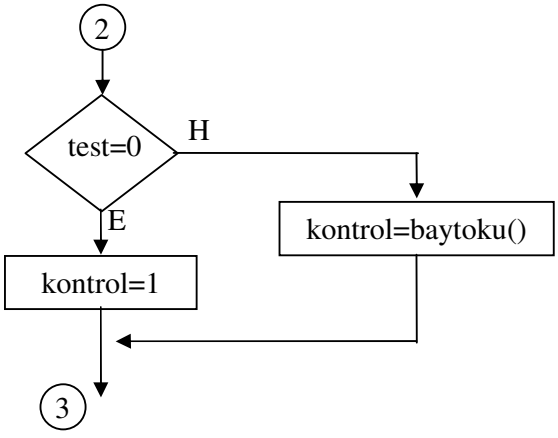
65535 değerinin aşılması durumunda, dosyadan ilk okunan bayt, tamamen kontrol amaçlı kullanılır. Bu baytın en anlamlı ilk biti daha önce de belirtildiği gibi, B kontrol değerinin 1 veya daha büyük bir sayı olduğunu belirlemede kullanılır. En anlamlı ikinci bitin kontrolü sonucu elde edilen 65535 sınırının aşılması durumunda, geri kalan bitler, dosyada bu sayıyı ifade etmek için kaç adet bayt bulunduğu bilgisini taşır. Bu bilgiler ışığında dosyadan gerekli sayıda bayt okunarak, bu değerler birleştirilir ve şifreli karakterin deşifrelenmesi için gerekli iterasyon sayısı elde edilir.

Sınır değerinin aşılması durumu, nadir rastlanan bir durumdur ancak kontrol edilmesi gerekir. Tüm bu işlemler, Şekil 27'de akış diyagramı ile açıklanmıştır.



Şekil 27. Taşma durumunda dosyadan bayt okuma

Son bölümde ise, b kontrol bitinin 1 değerli olup olmamasına göre dosyadan bir bayt daha okunup okunmayacağına bakılarak işlem sonlandırılır (Şekil 28). Tüm bu işlemler, dosyada tek bir bayt kalmayıncaya kadar devam ettirilir. Böylece, şifrenin çözülmesi için gerekli olan iterasyon sayısı ve b kontrol parametresinin değeri elde edilmiş olur. Bilindiği gibi b kontrol parametresi, iterasyon boyunca $C_i \oplus f_n(x) = \tilde{C}_i' \oplus f_n(x')$ durumunun kaç kez ortaya çıktığını belirtmede kullanılır.

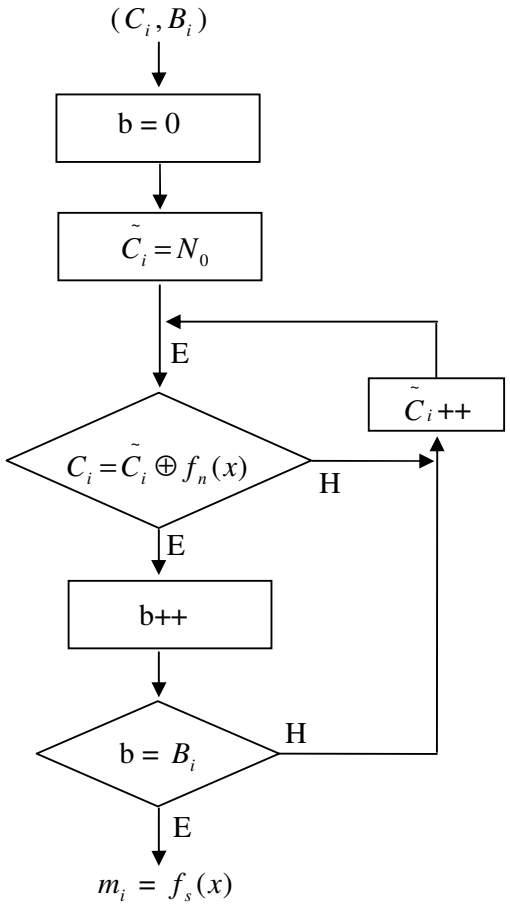


Şekil 28. Varsa, B kontrol baytının okunması

Dosyadan okunan değerler, iterasyon ve kontrol değeridir. Kontrol değeri, genellikle 1 dir. Çünkü, kontrol ettiği durum, çok düşük bir olasılıkla gerçekleşmektedir. Konrol parametresi ve başlangıç koşulu aynı olacak şekilde, şifrelemede kurulan sistemin bir benzeri oluşturulur.

Deşifreleme işlemi, şifrelemeden daha kısa sürede gerçekleşmektedir. Çünkü şifrelemede, her karakter için b kontrol parametresinin belirlenmesinde kullanılan büyük boyutlu dizinin elemanlarının sıfırlanması gerekmektedir. Deşifrelemede ise, bu değer dosyadan okunur.

Deşifreleme işleminin anlatıldığı akış diyagramı Şekil 29'da gösterilmektedir.



Şekil 29. Deşifreleme işlemi

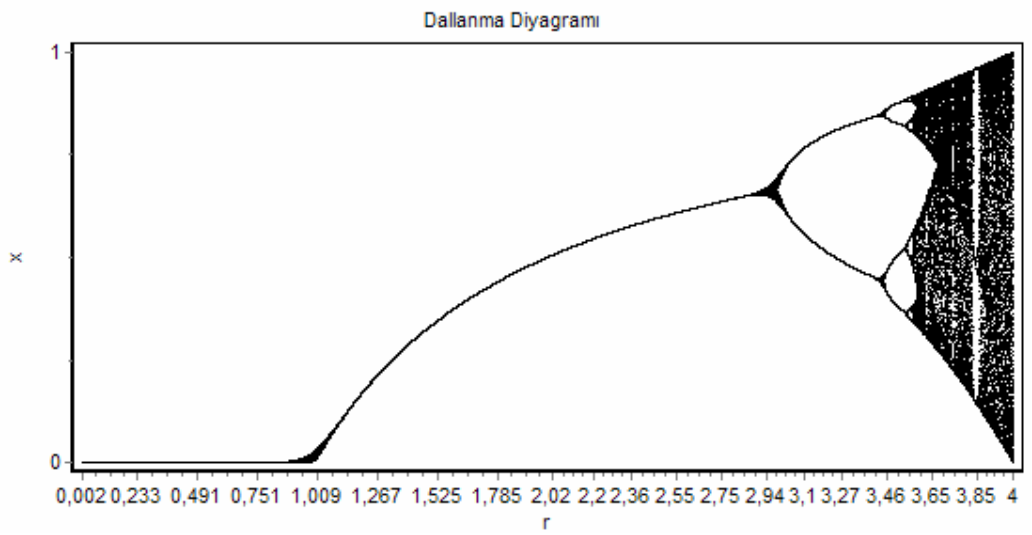
Karakterin deşifrelenmesi için gereken bilgiler elde edildikten sonra, aynı kontrol parametreleri kullanılarak uygun başlangıç koşullarıyla $C_i = \tilde{C}_i \oplus f_n(x)$ durumu

sağlanıncaya kadar, lojistik haritanın iterasyonu devam ettirilir. Bu durum sağlandığında, kontrol değeri de göz önüne alınarak, gerekirse işlem devam ettirilir, ya da sonlandırılır. Tüm bu işlemler, bir diğer karakter için de gerçekleştirilir. İlk karakter hariç, her bir karakterin deşifrelenmesinde kullanılacak başlangıç koşulu, bir önceki karakterin şifresinin çözüldüğü durumda gelinen nokta olarak alınır.

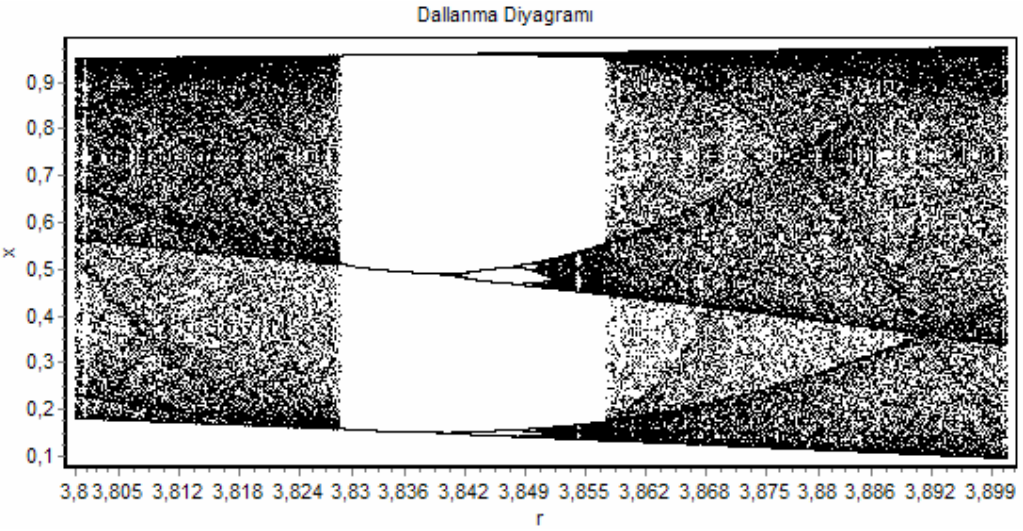
3. BULGULAR VE TARTIŞMA

3.1. Kaotik Davranışın Araştırılması

Kaotik bir sistemin farklı kontrol parametreleriyle iterasyona tabi tutulması sonucu, dallanma diyagramları oluşmaktadır. Bu dallanma diyagramı (Şekil 30), sistemin kaotikliği hakkında bir takım fikirler öne sürülmesine yardımcı olmaktadır.



Şekil 30. Lojistik haritaya ait dallanma diyagramı (r=[0,4])



Şekil 31. Lojistik haritaya ait dallanma diyagramı (r=[3.8,3.9])

Şekilde 30'daki dallanma diyagramının, r kontrol parametresinin 3.8 ile 3.9 arasındaki bölümü alınır ve yakınlaştırılırsa, Şekil 31'deki görüntü oluşur. Normalde diyagramda sonsuz adet nokta vardır. Ancak, diyagramın sağlıklı yorumlanabilmesi için şekil 30 ve Şekil 31, iterasyon sayısının sınırlı tutulmasıyla oluşturulmuştur.

Daha önceden de belirtildiği gibi, sistem kaotik özellik göstermeye başladıktan sonra, her zaman kaotik özellik sergilemez. Şekil 31'de de görüldüğü gibi, r kontrol parametresinin 3.83 değeri için, sistem üç noktalı bir çekici gibi davranır. Bunun anlamı şudur: Sistem, o kontrol değeri için iterasyona tabi tutulduğunda sürekli üç farklı noktaya gitmektedir. Farklı başlangıç noktaları alınarak yapılan iterasyonlar sonucu gelinen noktalar, Tablo 17'de verilmektedir.

Tablo 17. r=3.83 için iterasyon değerleri

Başlangıç noktası	**1. Nokta**	**2. Nokta**	**3. Nokta**
0.306	0.957416597518873	0.156149315683605	0.504666487408413
0.2	0.957416597518873	0.156149315683605	0.504666487408413
0.7777777	0.957416597518873	0.156149315683605	0.504666487408413

Tablodan da görüldüğü gibi, başlangıç noktaları farklı da olsa, sistem hep aynı konuma gider. Tabi ki, iterasyon başlar başlamaz üretilen değerler bunlar değildir. Kaotik sistemin, beklenen özellikleri gösterebilmesi için bir geçiş zamanı gerekmektedir ve her farklı başlangıç değeri ile yapılan iterasyonlarda, geçiş zamanı sonrasında varılan konumlar, Tablo 17'deki değerler olmaktadır. Başlangıç noktasının farklı olması, sistemin beklenen değerler üretmesini geciktirmekten başka bir sonuca neden olmaz.

Sistem, bu durumda kaotik özellik göstermez. Bu, Luapunov üstelinin -0.365196125101365 değerine sahip olmasından da anlaşılmaktadır.

Kontrol parametresinin 3.8515 olması durumunda sistem yine negatif bir değer üretmektedir: -0.00660725645825272. Bu kontrol parametresi ile sistem, 42 noktalı bir çekici gibi davranmaktadır. Bu davranışı 3530. iterasyon sonrasında istikrarlı olmaktadır. yani, peryodik değerlerin üretilmesi 3530. iterasyondan itibaren gerçekleşmektedir. Bu andan itibaren her 42 iterasyonda elde edilen değer, birbirinin aynıdır. Sistemin iterasyonlar sonucu geldiği noktalar tablosu Tablo 18'de gösterilmektedir.

Tablo 18. r=3.8515 için iterasyon değerleri

İterasyon Sonucu			
0,1433485213273	0,162568119837734	0,153421004959718	0,147686459466213
0,472963132212361	0,524342155652483	0,500244375258237	0,484808214006444
0,96005958346557	0,96059283010322	0,962874769991244	0,961986110922013
0,147686459466213	0,145795624068686	0,137679367547338	0,140844861513315
0,484808214006444	0,479662960163874	0,457264558940495	0,466060719399397
0,961986110922013	0,961282038128426	0,955840936521297	0,958438554332256
0,140844861513315	0,143348521327279	0,162568119837726	0,153421004959595
0,466060719399397	0,472963132212304	0,524342155652463	0,500244375257911
0,958438554332256	0,960059583465558	0,960592830103224	0,962874769991244
0,153421004959595	0,147686459466256	0,145795624068672	0,137679367547336
0,500244375257911	0,484808214006559	0,479662960163837	0,457264558940489
0,962874769991244	0,961986110922026	0,96128203812842	0,955840936521295
0,137679367547336	0,140844861513267	*0,1433485213273*	0,162568119837734
0,457264558940489	0,466060719399265	0,472963132212361	0,524342155652483
0,955840936521295	0,958438554332221	0,96005958346557	0,96059283010322

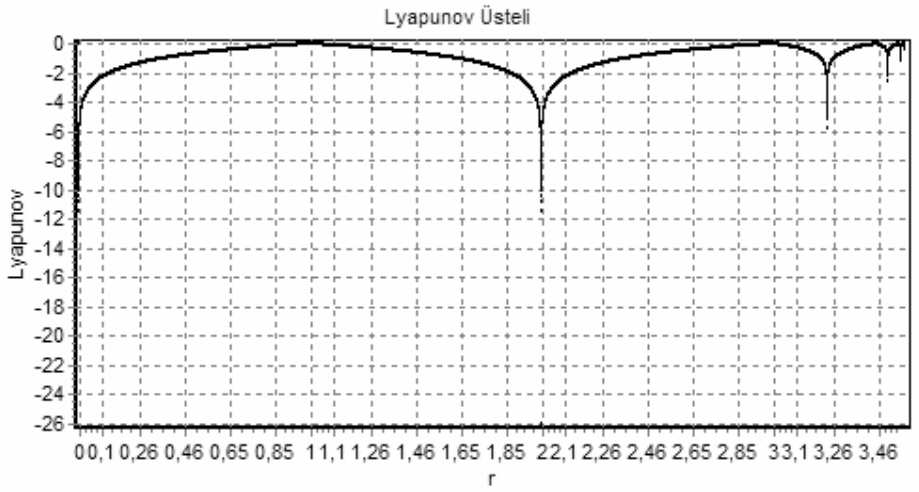
Kontrol parametresi olarak 3.87 değeri ile çalışması durumunda 0.426893432309443 pozitif Lyapunov üstelinin üretildiği ve kaotik davranışın sergilendiği durumda, iterasyonlar sonucu elde edilen değerlerin hiç biri birbirinin aynı olmayacaktır. Tablo 19'da, örnek olarak 4900. iterasyondan itibaren 60 adet iterasyon sonucu gösterilmektedir.

Tablo 19. r=3.87 için iterasyon değerleri

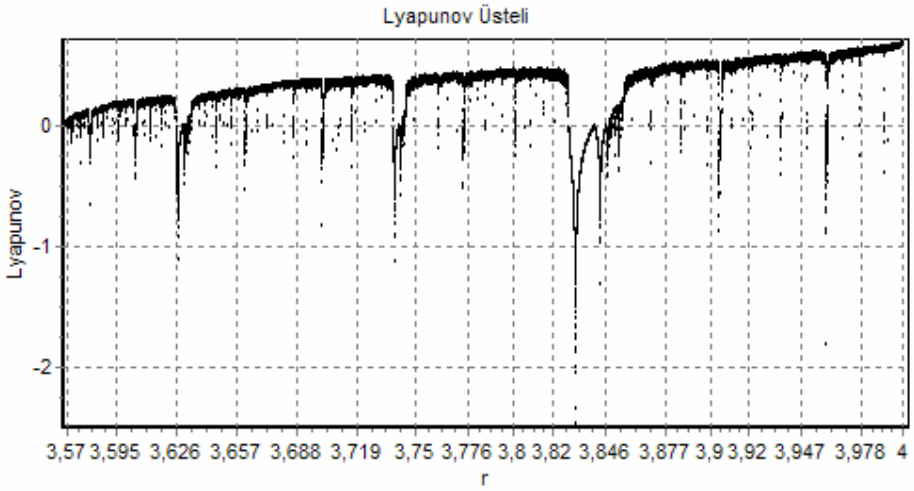
İterasyon Sonucu			
0,463630995979756	0,59012892958165	0,855119585448727	0,464670912733919
0,962381133765247	0,936063123323043	0,479454609718904	0,9626696811447
0,140108269223923	0,231615446088901	0,965866422450824	0,139075284998418
0,466249636000528	0,68874285982891	0,127588003795208	0,463368064890973
0,963091733038725	0,82963563418215	0,430767010670289	0,96230685214775
0,137563201081906	0,546987149381274	0,948950289639387	0,140374089149511
0,459135123477319	0,958955844158994	0,187476876864589	0,466989820427981
0,961037339424462	0,15232139314462	0,589514381347015	0,963282969532691
0,14491049231032	0,499692799116845	0,936490369309076	0,136877604845559
0,479537278716072	0,967499634778879	0,230172729526649	0,45721002815136
0,96587954213573	0,121688634033848	0,685737854701447	0,960414100856575
0,127540696089088	0,413627575173789	0,8339906089106	0,147132971684509
0,430630639019871	0,938629042368709	0,535802557127369	0,485626409468565
0,948877141100397	0,222929659548045	0,962539344615158	0,966700457594558
0,187731038204458	0,670407942329912	0,139541962623135	0,124577942749586

Burada iterasyonlar sonucu üretilen değerlerden örnek olarak, sadece 60 tanesi gösterilmektedir. Teorik olarak olaya yaklaşıldığında, sonsuz iterasyon yapıldığında üretilen değerlerin hiç biri birbirinin aynı olmaz. Ancak, günümüz bilgisayarlarının yapısı nedeniyle, bir kayan noktalı sayı, belirli bir değere yuvarlanacağından, kaosun bu ergodiklik özelliği pratikte biraz bozulmalar göstermektedir.

Lojistik haritada 0 ile 4 arasındaki kontrol parametreleri incelendiğinde, sistemin belirli kontrol parametresi ile çalışması durumunda kaotik özellik gösterdiği belirlenmiştir. Aşağıdaki şekillerde, 0 ile 4 arasında 10^{-5} adımla alınan kontrol değerlerine göre hesaplanan Lyapunov üstellerine ilişkin bir takım grafikler sunulmuştur.



Şekil 32. Lojistik haritanın r=[0,3.57] durumundaki Lyapunov üstelleri

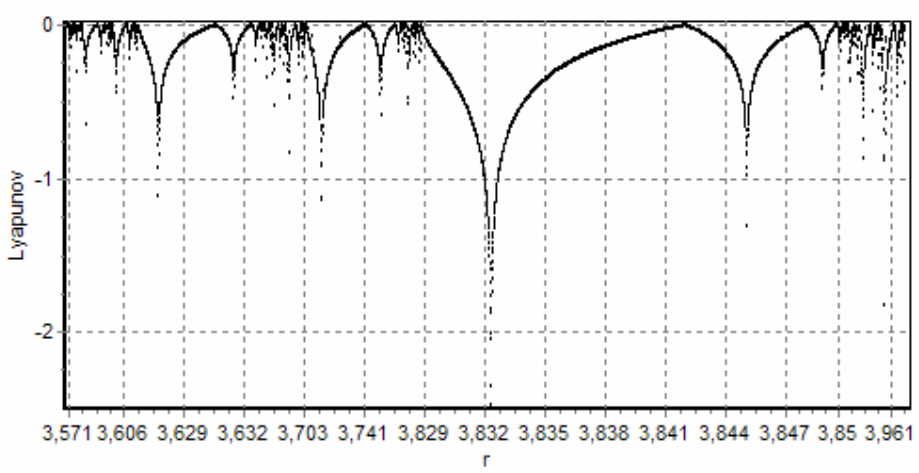


Şekil 33. Lojistik haritanın r=[3.57,4] durumundaki Lyapunov üstelleri

Lojistik harita, r kontrol parametresi 0 ile 3.56999…9 arasında olduğunda kaotik davranş sergilemez yani pozitif Lyapunov üsteline sahip değildir. Kaotik davranış özelliği 3.57 değerinden sonra başlar.

r kontrol parametresinin 3.57 ve 4 arası değerleri için de sistem her zaman kaotik davranış göstermez.

Şekil 34'te kontrol parametresinin 3.57 ve 4 arasındaki değerlerinin oluşturduğu negatif Lyapunov üstelleri gösterilmektedir. Bunun anlamı şudur: Sistem, bu değerlere sahip kontrol parametreleri ile çalıştırılırsa, kaotik davranış göstermeyecektir.



Şekil 34. Lojistik haritanın r=[3.57,4] durumundaki negatif Lyapunov üstelleri

Tüm bu değerlendirmeler göz önüne alınarak, kaotik özellik sağlayacak şekilde uygun kontrol parametreleri ile sistem çalıştırılarak istenen işlemler gerçekleştirilir.

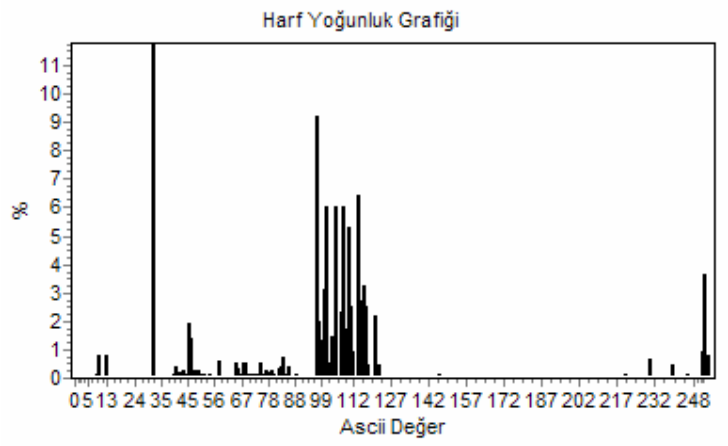
3.2. Oluşan Dosya Özelliklerinin Karşılaştırılması

Açık metin veya anahtarda yapılacak ufak değişikliklerin şifreli metindeki büyük etki yaratması olarak ifade edilen difüzyon ile, açık metnin istatistiklerinin, şifreli metnin istatistiklerinden bağımsız olması olarak ifade edilen konfüzyonun var olması, iyi bir kripto sistemin istatistiksel analizini önleyen önemli faktörlerdir.

Kaotik sistemlerin başlangıç koşullarına duyarlı olması difüzyona karşılık gelir ve bu sistemleri kullanılarak yapılan şifrelemede, açık metinle şifreli metin arasında herhangi bir ilişki yoktur. Bu da konfüzyon demektir. Geleneksel şifrelemede kullanılan algoritmalar, kripto sistemin difüzyon ve konfüzyon özelliklerini geliştirir. Ancak, kaotik şifreleme,

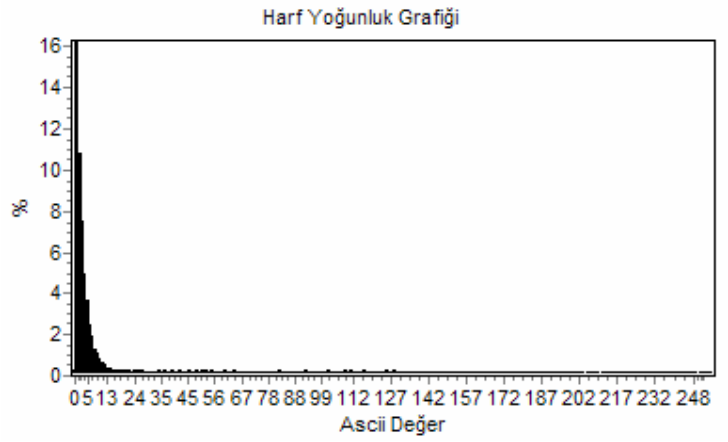
parametre uzayının büyük olması, kullanılabilecek pek çok eşleştirme fonksiyonu ve sonucunda oluşacak pek çok farklı şifre çeşitliliği nedeniyle geleneksel şifrelemede ortaya çıkan performansı aşan bir davranış sergilemektedir.

20.552 baytlık bir dosya, aşağıdaki şekildeki gibi bir harf yoğunluğuna sahiptir.



Şekil 35. Herhangi bir dosyanın harf yoğunluğu

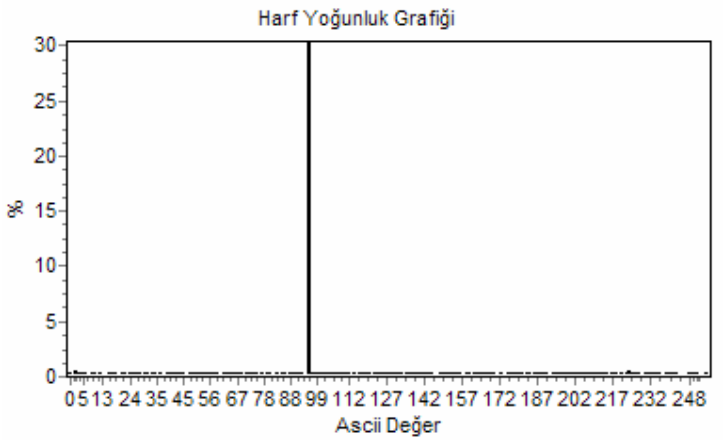
Kaotik şifreleme tekniği kullanıldığında, Şekil 36'daki yoğunluk grafiği oluşur.



Şekil 36. Şifrelemiş dosyanın yoğunluğu

Kullanılan algoritmada, metin içerisinde bulunan aynı harfler, sürekli, farklı şekilde şifrelendiğinden dolayı, oluşan şifreli metnin istatistiksel analizi zordur. Durumu daha da zor hale getirmek için, şifreli karakterler dosyaya yazılmadan sözde rastgele sayı üreten bir fonksiyon kullanılarak, daha da gizlenebilir. RC4 sözde rasgele sayı üreteci kullanılarak

maskeleme yapıldığında oluşan dağılım, şekildeki gibi olmaktadır. Bu durum konfüzyonu güçlendirirken, şifreleme hızını düşürmektedir.



Şekil 37. RC4 kullanılarak şifrelenmiş dosyanın yoğunluğu

3.3. Şifreleme Yöntemlerine Göre İşlem Süreleri ve Dosya Boyutları

Program, 1.86 GHz Intel Pentium işlemci ve 1 GB RAM belleğe sahip bir makinede çalıştırılmıştır. Her iki şifreleme yöntemi için, aynı text dosyalar kullanılması sonucunda, oluşan dosya boyutları (bayt cinsinden) ve şifreleme - deşifreleme hızları (sn cinsinden) Tablo 20 ve Tablo 21'de gösterilmektedir.

Tablo 20. Normal kaotik şifreleme

Dosya Boyutu	Şifreleme Hızı	Deşifreleme Hızı	Şifreli Dosya Boyutu
20.285	1.11	0.453	40.482
32.662	1.563	0.719	64622
88.802	4.438	1.64	176.292
113.060	5.593	2.141	225.334
524.420	27.96	10.093	1.028.148

Tablo 21. RC4 sözde rasgele sayı üreteci kullanılarak yapılan kaotik şifreleme

Dosya Boyutu	Şifreleme Hızı	Deşifreleme Hızı	Oluşan Dosya Boyutu
20.285	93.687	61.594	40.482
32.662	113.547	95.547	64.623
88.802	304.281	266.891	176.295
113.060	393.062	341.687	225.335
524.420	1879.5	1613.65	1.028.161

4. SONUÇLAR

Bilgi güvenliğini sağlama, geçmişten bu güne kadar her zaman için gerekli görülmüştür. Günümüzde, teknolojinin oldukça gelişmiş olması ve tüm bilgi alış verişinin bilgisayar ağları üzerinden yapılıyor olması, veri güvenliğinin önemini artırmaktadır. Bunu sağlamak amacıyla verilerin şifrelenmesi gerekmektedir. Bu amaçla, geliştirilen bir çok şifreleme tekniği bulunmaktadır.

İki kaotik sistemin senkron çalışabileceği yaklaşımından yola çıkılarak, çalışmada, güvenli veri iletişiminin sağlanabilmesi amacıyla, kaotik sistemler kullanılarak, veri şifrelemesi gerçekleştirilmiştir. Kripto sistem, kaotik sistemlerin genel özelliklerinden yararlanılarak elde edilen şifreli verinin, gönderici konumundaki kaotik sistemin alıcı tarafla düzenli aralıklarla senkronize olması sonucu, alıcı tarafa iletilip, burada benzer işlemlerle çözülmesi esasından oluşur.

Çalışma esnasında ilk olarak, veri şifreleme teknikleri incelenmiş, kaotik sistemlerin veri şifrelemesi için uygun özellikleri belirlenerek, bu özelliklerden yararlanılarak verilerin şifreleme ve deşifrelemesi gerçekleştirilmiştir.

Haberleşme kısmı incelenmemiş olup, verinin şifrelenmesi ve deşifrelenmesi işlemi üzerinde çalışılmıştır. Bunun için iki farklı yöntem kullanılmıştır. Birincisinde, kaotik sistemlerin bir takım özellikleri kullanılarak şifreleme gerçekleştirilmiş, ikincisinde de, bir önceki yöntemde üretilen sonuçlar, kripto analizin daha da zorlaştırılması için, RC4 yalancı rasgele sayı üreteci kullanılarak maskelenmiştir.

Her iki yöntemde de elde edilen sonuçlar karşılaştırılmış ve performans değerlendirmesi yapılmıştır.

5. ÖNERİLER

1. Çalışmada, senkron çalışan ve kaotik özellik gösteren iki sistemin, veri şifrelemede kullanılabileceği önerilmektedir. Kaotik sistemler kullanılarak, verinin matematiksel yollarla şifrelenip karşı tarafa gönderilmesiyle, güvenli haberleşme yapmak mümkündür.

2. Şifreli metnin transferi esnasında, sadece başlangıçta senkronizasyon yapılmasının yeterli olmayacağı anlaşılmıştır. Çünkü kaotik sistemlerin karakteristik özelliği nedeniyle belirli bir süre sonra iletişim kopacaktır. Bunu önlemek için, düzenli aralıklarla senkronizasyonun sağlanması önerilmektedir.

3. Baptista türü şifreleme tekniğinin, kaotik sistemler aracılığıyla güvenli veri transferinde yeterli olduğu görülmüştür. Ancak, yöntem daha da hızlandırılıp, şifreli metnin boyutu daha da küçültülerek, teknik daha da verimli hale getirilebilir.

4. Tek boyutlu kaotik şifrelemenin yanı sıra, mevcut kaotik sistemler iki katmanlı kullanılarak, daha da güçlü bir şifreleme gerçekleştirilebilir.

5. Kaotik şifreleme her ne kadar kriptoanaliz yaklaşımlara karşı dirençli olsa da, mümkün olabilecek açıkların kapatılması, oluşan şifrelerin, uygun fonksiyonlarla ayrıca işleme sokulması ile sağlanabilir.

6. Gelişen teknoloji ile birlikte, bilgi güvenliği daha önemli hale gelmiştir. Mevcut şifreleme tekniklerinin, bu ilerleme karşısında yetersiz kalabileceği dikkate alınarak, daha güçlü algoritmaların geliştirilmesi gerekmektedir.

6. KAYNAKLAR

1. http://www.btguvenlik.telekom.gov.tr, Bilgi Güvenliği, 13 Mayıs 2006.

2. http://www.byte.com.tr, Dosya Şifreleme Programları, 01 Şubat 2006.

3. Menezes, A., Van Oorschot, P., C. and Vanstone, S., A., Handbook of Applied Cryptography, CRC Press, October 1996.

4. http://www.mutasyon.net, Kriptografiye Giriş, 12 Ocak 2006.

5. Bozkurt, F., Elektronik Güvenlik, Şifreleme Teknikleri ve Algoritması Açık Olan Şifreleme Teknikleri (Public Key Encryption), İzmir, 02 Kasım 2005.

6. Akyıldız, E., Doğanaksoy, A., Keyman, E., Uğuz, M., Kriptografi Ders Notları, ODTÜ Uygulamalı Matematik Enstitüsü, Şubat 2004.

7. Eren, A., M., Açık Anahtarlı Kriptografi, Penguence, Mart 2005, 28-32.

8. http://www.simetri.com, Sayısal İmza Nedir?, 01 Mayıs 2006.

9. Briggs J. ve Peat, F., D., Kaos ve Yedi Yaşam Dersi, Ege Meta Yayınları, İzmir, 2001.

10. Gleick, J., Kaos, Tübitak Popüler Bilim Kitapları, Ankara, 1995.

11. Elert, G., The Chaos Hypertextbook, http://hypertextbook.com/chaos, 10 Eylül 2005.

12. Clayton, K., Basic Concepts in Nonlinear Dynamics and Chaos, Society for Chaos Theory in Psychology and Life Sciences Meeting, Miwaukee, USA, 1997.

13. Li, S., When Chaos Meet Computers, Elsevier Science, June 2004.

14. Pecora, L., M. and Carroll, T., L., Synchronization in Chaotic Systems, Physical Review Letters, Vol. 64, No. 8, February 1990.

15. Carroll, T., L. and Pecora, L., M., Synchronizing Chaotic Circuits, IEEE Transactions on Circuits and Systems, Vol.38, No. 4, April 1991.

16. Li, Z., Li, K., Wen, C. and Soh, Y., C., A New Chaotic Secure Communication System, IEEE Transactions on Communications, Vol.51, No.8, August 2003.

17. Kocarev, L., Chaos Based Cryptography: A Brief Overview, IEEE Circuits and Systems Magazine, Vol.1(3), 2001, 6-21.

18. Alvarez, G. and Li, S., Some Basic Cryptographic Requirements for Chaos-Based Cryptosystems, International Journal of Bifurcation and Chaos, Vol.16, No. 7, May 2005.

19. Yang, T., A Survey Of Chaotic Secure Communication Systems, International Journal Of Computational Cognition, Vol.2, No.2, June 2004, 81-130.

20. Kangwei, H. and Lih, T., C.,Parwani, Chaos and Cryptography: Applications and Analysis, Singapore, 2003.

21. Baptista, M., S., Cryptography with Chaos, Physics Letters A 240, 1998, 50-54.

22. Li, S., Baptista-Type Chaotic Cryptosystems: Problems and Countermeasures, Physics Letters A 332, September 2004, 368-375.

23. Jenkins, R., J., Fast Software Encryption, Third International Workshop, Lecture Notes in Computer Science, Springer, Vol.1039, 1996, 21-23.

24. Konheim, A., Cryptography: A Primer, New York, Wiley, 1981.

25. Feistel H., Cryptography and Computer Privacy, Scientific American, Vol.228, May 1973, 15-23.

26. Solak, E., Cryptanalysis of Observer Based Discrete-Time Chaotic Encryption Schemes, International Journal of Bifurcation andChaos, Vol.15, No.2, 2005.

27. Solak, E., On the Security of Discrete-Time Chaotic Cryptosystems, Physics Letters A, Vol.320, No.5-6, 2004, 389-395.

28. Yang, T. and Chua, L., O., Secure Communication via Chaotic Parameter Modulation, IEEE Transactions, Vol.43, 1996, 817-819.

29. Kolumban, G., Kennedy, M., P. And Chua, L., The Role of Synchronization in Digital Communication Using Chaos-Part II: Chaotic Modulation and Chaotic Synchronization, IEEE, Vol.45, 1998, 1129-1140.

ÖZGEÇMİŞ

1981 yılında Trabzon'un Maçka ilçesinde doğdu. İlköğretimini Fatih İlkokulu, orta öğretimini Cumhuriyet Ortaokulu ve Fatih Lisesi'nin Yabancı Dil Ağırlıklı bölümünde tamamladı. 1999 yılında Karadeniz Teknik Üniversitesi, Mühendislik Mimarlık Fakültesi, Bilgisayar Mühendisliği Bölümü'nde okumaya hak kazandı. Bu bölümden, 2003 yılında bölüm ikincisi olarak mezun oldu. Aynı yıl, Karadeniz Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Ana Bilim Dalı'nda Yüksek Lisans Programına başladı. 2005 yılından itibaren, Türk Telekom A.Ş. Genel Müdürlüğünde Uzman Yardımcısı olarak görev yapmaktadır.