KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

DNA DESENLERI KULLANARAK GÖRÜNTÜ ŞİFRELEME

YÜKSEK LİSANS TEZİ

Sahereh HOSSEİN POUR

ARALIK 2011
TRABZON

KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

DNA DESENLERI KULLANARAK GÖRÜNTÜ ŞİFRELEME

Sahereh HOSSEİN POUR

Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsünce
"YÜKSEK LİSANS (BİLGİSAYAR MÜHENDİSLİĞİ)"
Unvanı Verilmesi İçin Kabul Edilen Tezdir.

Tezin Enstitüye Verildiği Tarih : 06.12.2011
Tezin Savunma Tarihi : 21.12.2011

Tez Danışmanı : Yrd.Doç. Dr. Hüseyin PEHLİVAN

Trabzon 2011

Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalında

Sahereh Hossein Pour tarafından hazırlanan

DNA DESENLERI KULLANARAK
GÖRÜNTÜ ŞİFRELEME

başlıklı bu çalışma, Enstitü Yönetim Kurulunun 06 / 12 / 2011 gün ve 1432 sayılı kararıyla oluşturulan jüri tarafından yapılan sınavda

YÜKSEK LİSANS TEZİ

olarak kabul edilmiştir.

Jüri Üyeleri

Başkan : Yrd.Doç. Dr. Hüseyin PEHLİVAN .....

Üye : Yrd.Doç. Dr. Mustafa ULUTAŞ .....

Üye : Yrd.Doç. Dr. H.İbrahim OKUMUŞ .....

Prof. Dr. Sadettin KORKMAZ

Enstitü Müdürü

ÖNSÖZ

"**DNA Desenleri Kullanarak Görüntü Şifreleme** " adlı bu çalısma, Karadeniz Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisligi Anabilim Dalında Yüksek Lisans Tezi olarak hazırlanmıştır.

Çalışma süresince bilgi, görüş, öneri ve yardımlarını esirgemeyen saygıdeğer danışmanım Yrd.Doç. Dr. Hüseyin PEHLİVAN'a sonsuz teşekkürlerimi sunarım. Ayrıca yardımlarından dolayı Prof. Dr. Rıfat YAZICI'ya , çalısmam boyunca manevi desteklerini eksik etmeyen eşim Mir Mohammad Reza ALAVİ MİLANİ'ye ve tüm çalısma arkadaslarıma teşekkür ederim.

Tüm eğtim-öğretim hayatım boyunca maddi ve manevi desteklerini eksik etmeyen aileme sonsuz saygı, sevgi ve teşekkürlerimi sunarım. Bu tezin, bundan sonraki çalışmalara katkı sağlamasını temenni ederim.

Sahereh HOSSEİN POUR
Trabzon 2011

TEZ BEYANNAMESİ

Yüksek Lisans Tezi olarak sunduğum "DNA Desenleri Kullanarak Görüntü Şifreleme" başlıklı bu çalışmayı baştan sona kadar danışmanım Yrd.Doç. Dr. Hüseyin PEHLİVAN 'ın sorumluluğunda tamamladığımı, verileri/örnekleri kendim topladığımı, deneyleri/analizleri ilgili laboratuarlarda yaptığımı/yaptırdığımı, başka kaynaklardan aldığım bilgileri metinde ve kaynakçada eksiksiz olarak gösterdiğimi, çalışma sürecinde bilimsel araştırma ve etik kurallara uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ettiğimi beyan ederim. 07/12/2011

Sahereh HOSSEİN POUR

İÇİNDEKİLER

Sayfa No

Yüksek Lisans Tezi

ÖZET

DNA DESENLERİNİ KULLANARAK GÖRÜNTÜ ŞİFRELEME

Sahereh HOSSEİN POUR

Karadeniz Teknik Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Ana Bilim Dalı
Danışman : Yrd. Doç. Dr. Hüseyin PEHLİVAN
2011, 107 sayfa

Günümüz bilgisayar destekli şifreleme teknikleri oldukça yüksek düzeyli bilgi gerektiren karmaşık güvenlik önlemleriyle yoğrulmuş teknikler içerir. Öncekilerden daha güvenli olduğu sanılan her bir yeni tekniğin zaman içerisinde başka güvenlik açıklarının bulunduğuna şahit olmaktayız. Dolayısıyla, temel ilke olarak herhangi bir şifreleme yönteminin kırılamaz olmadığını ve sonlu bir süre sonunda şifresinin çözülebileceğini söyleyebiliriz. Görüntü verilerine uygulanabilen şifreleme yöntemlerinin sayısı da günden güne artmaktadır. Bu çalışmada, Henon kaotik sistemleri ile lojistik haritanın rastgele özelliklerinden yararlanılarak, DNA desenlerinden kullanıp görüntü şifrelemede kullanılabilecek hızlı bir algoritma geliştirilmiştir. Geliştirilmiş algoritma daha sonra bir Turing Makine üzerinde modellenmiştir. Siyah-beyaz ve renkli resimler üzerindeki uygulamalardan elde edilen sonuçlar algoritma güvenliğinin yüksek olduğunu göstermektedir.

Anahtar Kelimeler: Kaos, Görüntü Şifreleme, Lojistik Harita, Rasgele , DNA Desenleri, Turing Makinası.

Master Thesis

SUMMARY

IMAGE ENCRYPTION USING BY DNA PATERNS

Sahereh HOSSEIN POUR

Karadenik Technical University
The Graduate School of Natural and Applied Sciences
Computer Engineer Graduate Program
Supervisor: Assist.Prof.Dr. Hüseyin PEHLİVAN
2011, 107 Pages

Today's computer-aided encryption techniques requires knowledge of very complicated and complex security measures. While each claims to be more secure than the previous, with every coming days we are witnessing how the previous passwords are broken. Therefore, based on the basic principles learned in theory, it is possible to say that any encryption method cannot become "unbreakable" and any password can break in a limited period of time. The number of encryption methods on images is increasing gradually. In this study, by means of a logistic map, we propose a simple and fast encryption algorithm to encrypt the images, using the Henon chaotic systems, logistic properties of random maps and DNA paterns. Also we modeled our algorithms with Turing machines. This algorithm is applied to both the black-and-white and color images. The results indicate a greater security of the proposed algorithm.

Key Words: Chaos, image encryption, logistic map, random numbers , DNA Patterns, Turing Machines.

ŞEKİLLER DİZİNİ

Sayfa No

TABLOLAR DİZİNİ

Sayfa No

1. GENEL BİLGİLER

1.1. Giriş

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

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

Bilgi güvenliği; başkası tarafından dinlenme, bilginin değiştirilmesi, kimlik taklıdı gibi tehditlerin ortadan kaldırılması ile sağlanır ve bu amaçla kullanılan temel araç şifrelemedir. Ağ ve haberleşme güvenliği için en önemli araç şifrelemedir. Kriptografi bilgi güvenliğini inceleyen ve anlaşılabileni anlaşılamaz yapan bir bilim dalıdır.

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

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

Ticari işlerde, devlet işlerinde, askeri işlerde, personel ilişkilerinde güvenli is 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.

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

Kriptografi genel olarak şu ana konularla ilgilenir:

- Gizlilik: Bilgi istenmeyen kişiler tarafından anlaşılamamali.
- Bütünlük: Bir iletinin alıcısi bu iletinin iletim sirasında değişiklige ugrayip uğramadiğini ögrenmek isteyebilir; davetsiz bir misafir doğru iletinin yerine yanliş bir ileti koyma şansına erişmemelidir. Saklanan veya iletilmek istenen bilgi farkina varılmadan değistirilememeli.
- Reddedilemezlik: Bilgiyi gönderen kişi, daha sonra bilgiyi kendisinin gönderdiğini inkar edememeli.
- Kimlik belirleme: Gönderen ve alıcı, birbirlerinin kimliklerini doğrulayabilirler. Davetsiz bir misafir başkasının kimliğine bürünme şansına erişmemelidir.

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 varılmiştır. Çalışmalar, yüz yıl öncesinden başlamiş olmasına ragmen, 90'lı yıllardan itibaren, kaotik sistemlerin senkron davranış göstereceğini kanıtlayan çalışmalar sonrasında hız kazanmıştır.

Bu çalışmadaki amaç, kaotik özellik gösteren sistemlerin bu özelliğinden yaralanılarak, ve DNA ilkelerin yarardımı ile verilerin şifrelenme ve deşifrelenmesini gerçekleştirmektir. Bu amaçla, lojistik harita incelenmiş, DNA özellikleri göz önüne alınmış ve şifreleme işlemlerin yapmak amacıyla bir Turing makinesi tasarlanmıştır. kaotik davranış gösteren bölümlerinden yararlanılarak şifreleme işlemlerin tersi uygulanarak da deşifreleme işlemi gerçekleştirilmiştir.

1.2. Kriptografinin Tarihi

Heredot'un anlattıklarına göre eski Yunan'da şifreli bir mesaj gönderilmek istendiğinde, kölelerin kafa derisi üzerinde mesajlar aktarılmaktaydı. Önce bir kölenin kafası tıras 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 tıras 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 degnek kullanmiştır. Sezar, şerit halindeki kağıdı degnek etrafına sarar ve mesajını boylamasına bu kağıt üzerine yazardı. Daha sonra kağıdın kıvrımlarını açarak düz hale getirir ve gönderecegi adrese yollardı. Sonraları Roma orduları iletişimlerinde yine Sezar'ın bulduğu alfabede 3 harf kaydırma şifrelemesini kullanmişlardır. Bu yöntemde, örneğin "A" harfi yerine "D", "B" harfi yerine "E" kullanmaktaydı. Oldukça basit ve hedefine ulasan bu yöntem o çagın 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 sağlamiştır. 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 mesajlasmalar, güvenli bir iletişimi oluştururdu.
2. Dünya savaşında ise filmlere konu olan Enigma makinesi Almanların en güvendiği şifreleme tekniğiydi, ta ki; Ruslara esir düsen 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 sansına 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 çıkarılmiştır. 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 [1].

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

1.3. Bilgisayar ve Güvenlik

Kitlesel iletişim araçlarının önemini daha çok arttırdığı günümüz dünyasında bilgiye hızlı bir şekilde ulaşmak gelişmiş toplumların temel ihtiyaçlarından birisi olmuştur. Bu ihtiyaç sayesinde bu toplumlarda 1980'li yıllardan itibaren bilgisayar ağları konusunda önemli gelişmeler sağlanmıştır. Internet te, bu çalışmalar neticesinde ortaya çıkan ve yaygın olarak kullanılan bir bilgisayar ağıdır.

Bütün bilgileri hızlı bir şekilde elde etmek için ilk önce kuruluşlar kendi yerel ağlarını kurmuşve daha geniş bir etkinlik alanına sahip olabilmek için yerel ağlarını dünyaya entegre etmek ihtiyacını hissetmişlerdir. NetWork teknolojisi de bununla birlikte gelişmiş ve değişik protokollerin ortaya çıkması kaçınılmaz olmuştur. Bilginin bir varlık olarak hasarlardan korunması, doğru teknolojinin, doğru amaçla ve doğru şekilde kullanılarak bilginin her türlü ortamda, istenmeyen kişiler tarafından elde edilmesini önlemektir. Buna uygun tanımı: elektronik ortamlarda verilerin veya bilgilerin saklanması ve taşınması esnasında bilgilerin bütünlüğü bozulmadan, izinsiz erişimlerden

korunması için, güvenli bir bilgi işleme platformu oluşturma çabalarının tümüdür. Bunun sağlanması için duruma uygun güvenlik politikasının belirlenmesi ve uygulanması gereklidir.

Internet sınırsız bir bilgi ortamı olmasına rağmen yasalarla tam anlamıyla denetlenememektedir. Kusursuz olmayan NetWork ağlarında güvenlik açıkları bulunmaktadır. Bazı kullanıcılar bu güvenlik açıklarından faydalanarak bazı sistemlere girip onlara zarar verebilirler. Bu da NetWork ortamında güvenliği sağlamak amacına yönelik yazılımların ve sistemlerin ortaya çıkmasına sebep olmuştur. Bilgisayar teknolojilerinde yer alan bilgisayar güvenliğinin amacı ise: "kişi ve kurumların bu teknolojilerini kullanırken karşılaşabilecekleri tehdit ve tehlikelerin analizlerinin yapılarak gerekli önlemlerin önceden alınmasıdır". Bilgi ve bilgisayar güvenliği daha genel anlamda, güvenlik konularını detaylı olarak ele alan "güvenlik mühendisliği"nin bir alt alanı olarak görülmektedir.

1.4. 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 bütünlügü, 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.

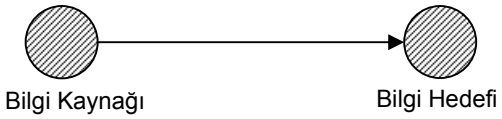
İşletmeler giderek sahip oldukları bilgi sistemleri ve ağları bilgisayar destekli sahtekarlık, casusluk, sabotaj, yıkıcılık, yangın ve sel gibi çok geniş kaynaklardan gelen tehdit ve tehlikelerle karşı karşıyadırlar. Bilgisayar virüsleri, bilgisayar korsanları ve hizmet saldırıları gibi yıkıcı kaynaklar daha yaygın, daha hırslı ve daha karmaşık hale gelmeye başlamiştır. 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. Dagıtılmiş bilgi işlemeye olan egilim, merkezi, uzman denetimin etkinliğini zayıflatmiştır. Bilgi sistemleri henüz yeterli güvenlik seviyesinde tasarlanmamiştır. Teknik olanaklar aracılığıyla ulaşılabilen güvenlik sınırlıdır, uygun yönetim ve yöntemlerle desteklenmelidir. Hanği 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 [2].

1.4.1. Bilgi Tehditleri

Kriptografi temelde bazı ana konulara yönelir. Bu alanlardan birincisi gizliliktir. Bilgi kesinlikle istenmeyen kişilerin eline geçememelidir. Bir diğeri ise bütünlüktür. Gönderilen bilgi bir bütün halinde olmalıdır, davetsiz misafirler doğru bilgiyi yanlış bir bilgi ile değiştirme imkanına sahip olmamalıdırlar. Bilgi gönderen ya da hazırlayan daha sonra, bunu kendisinin gönderdiğini rededememelidir. Son olarak gönderen ve alıcı birbirlerinin kimliklerini doğrulayabilmelidirler. Davetsiz bir misafir başka birinin kimliğine bürünememelidir.

Genelde bilgiler bilgi kaynağından, bilgi hedefine erişimi beklenen durumdur (Şekil 1).



Şekil 1. Bilginin beklenen erişimi

Ancak bazı durumlarda bazı yalnışlıklar olabilir. Bu durumda saldırıcıların davranışların aşağıdakı gibi sınıflandırabiliriz :

1.4.1.1 kesme (interrupt)

1.4.1.2 durdurma(interception)

1.4.1.3 modifikasyon (modification)

1.4.1.4 imalat (manufacturing produce)

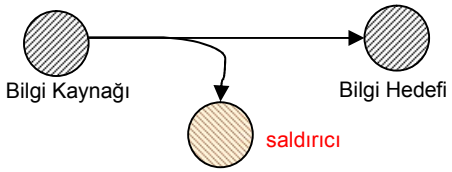
1.4.1.1. Kesme (interrupt)

Sistem kiymetli varlığı (asset) çöktürülür,ve ya hazır olmaktan çıkarılır ve ya kullanılmaz duruma sokulorç bu varlığa ataktır. Örnek olarak harddisk parçasının çökölmesi heberleşme hattının kesilmesi veya dosya yönetim sistemının yetkisiz kalmasını sağlar. Kesme atağı şekil 2 de gösterilmiştir.

Şekil 2. Bilgilerin erişiminde Kesme saldırısı

1.4.1.2. Durdurma (interception)

Otorizesiz (authorizesiz) partı kiymetli bir varlığa erişim kazanır. Bu durum gizliliğe (confidentiality) ataktır.

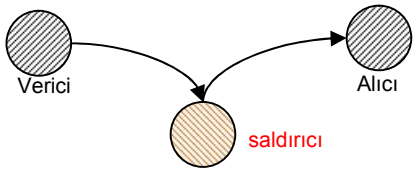


Şekil 3. Bilgilerin erişiminde durdurma saldırısı

Otorizesiz partı bir şahıs, bir program veya bir bilgisayar olabilir bilgi alıcıya gidiyor ama 3.kişi bilgiyi kendine kopyalıyor. **Örnek:** Ağdaki datayı elde etmek için kanalı dinlemeyi (wiretapping) ve dosyalar veya programların kanun'suz (illicit) kopyalanmasını sağlar.

1.4.1.3. Modifikasyon

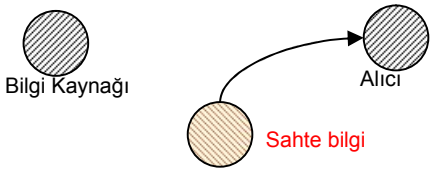
Otorizesiz partı erişimi ele geçirmekle kalmaz aynı zamanda varlıkları'da kurcalar (tamper).bu bütünlüğe ataktır.data dosyalarındaki dataların değiştirilmesi farklı gerçeklemeler yapacak şekilde bir program değiştirmek ve ağdan gönderilen mesajların içeriklerini değiştirmeyi sağlar.



Şekil 4. Bilgilerin erişiminde modifikasyon saldırısı

1.4.1.4. İmalat (fabrication)

Bu saldırıda , otorizesiz partı sisteme sahte nesneler üretip sokar. Bu ise hakıkılığe (Authenticity) bir attaktır.

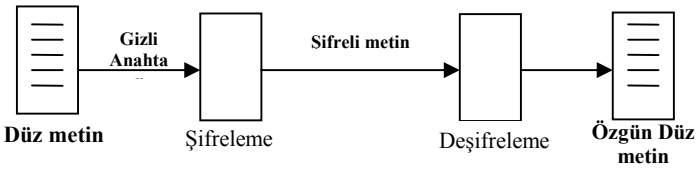


Şekil 5. Bilgilerin erişiminde imalat saldırısı

1.4.2. Şifrelemenin Temel Elemanları

Elektronik iletişim, günümüzde kağıt üzerinde yazı yazarak yapılan her türlü iletişimin yerine geçmeye başlamıştır. Kişi/kuruluş/toplumların, özel/kamusal/resmi haberleşmelerini elektronik iletişim ağları üzerinden yapabilmeleri, açık ağlar üzerinden iletilen bilginin güvenliği ve güvenilirliğiyle yakından ilgilidir. Bir göndericinin, bir alıcıya açık ağlar üzerinden bir ileti gönderdiği zaman, açık ağlardan gönderilen iletiler üçüncü şahıslar tarafından dinlenme ve değiştirilme tehdidi altindadırlar .

Burada söz konusu ileti düz metindir. Bazı kullanımlarda *plaintext* adi da verilir. Bir iletinin içeriğini saklamak üzere yapılan gizleme işlemi de şifrelemedir (*Encryption*). Bu işlem düz metini şifreli metine dönüştürür. Bilginin içeriği başkalarının anlamayacağı hale gelir. Bu bilgi, bir yere iletilmek amacıyla şifrelenen bir mesaj veya saklanmak amacıyla şifrelenen bir bilgi olabilir. Şifrelenmiş bir ileti şifreli metindir (*Ciphertext*). Şifreli metini düz metine geri çevirme işlemi şifre çözümüdür (*Decrypt*). Bu işlemler Şekil 6'de gösterilmektedir.



Şekil 6. Şifreleme ve şifreyi çözme işlemleri

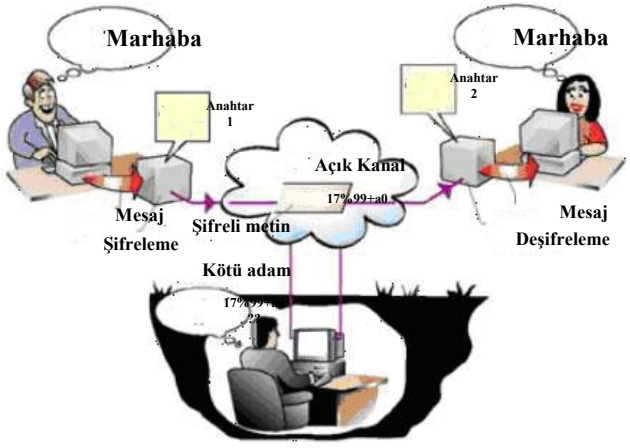
İleti güvenliğini sağlama bilimi *Kriptografidir* . Matematiğin hem şifre bilimi hem de şifre analizini kapsayan dalı *Kriptolojidir* ve şifre bilimciler tarafından icra edilir. Düz metin bir metin dosyası, resim, ses, görüntü dosyası yani sayısal ortamındakı her şey olabilir. Bilgisayar için düz metin sadece ikili (binary) bir veridir. Düz metin iletim veya depolama amacıyla tasarlanabilir. Her iki durumda da, düz metin şifrelenecek iletidir. Şifreli metin de ikili veridir, bazen düz metin ile ayni boyutta, bazen daha büyüktür. Şifreleme işlemi şifreli metini üretmek üzere düz metin üzerinde gerçekleştirilir.

Kriptografik algoritmalar adi da verilen şifreleme algoritmaları, şifreleme ve şifre çözümü için kullanılan matematiksel işlemlerdir. Eğer bir algoritmanın güvenliği bu algoritmanın çalışma biçimini gizlemeye dayalıysa, bu bir sinirlandırılmış algoritmadır. Geçmişte ilgilenilen kriptografi algoritmaları algoritmanın gizliliğine dayanmaktaydi. Sinirlandırılmış algoritmalar günümüzün şartlarına pek uymamaktadır; bir gruba ait kullanıcılar bunları kullanamamaktadır, çünkü gruptan bir kullanıcının her çikişinda geri kalan herkesin başka bir algoritmaya geçmesi gerekmektedir. Içlerinden birisi yanlişlikla ya da bilinçli olarak kullanılan algoritmayı açığa vurduğunda, diğer herkesin algoritmalarını değiştirmeleri gerekmektedir. Daha da kötüsü, sinirlandırılmış algoritmalar kalite kontrolüne ve standardizasyona olanak tanımamaktadır. Her bir grup kullanıcının

kendisine ait bir algoritması olmalidir. Bu tür bir grup hazır şifre çözüm anahtarının yazılım veya donanim ürünlerini kullanamaz; davetsiz bir misafir ayni ürünü alip algoritmayı ögrenebilir. Kendi algoritmalarını ve gerçekleştirimlerini kendileri yazmaları gerekir. Bu tip ciddi eksiklikleri olmasına rağmen, sinirlandırılmış algoritmalar düşük güvenlik gerektiren uygulamalarda sik sik kullanılmaktadır. Kullanıcılar sistemlerinde bulunan güvenlik sorunlarının ya farkinda değildir ya da bunları önemsememektedir.

Günümüz kriptografisi bu sorunu bir anahtar ile çözmektedir. Bu anahtar çok çeşitli değerler alabilen herhangı bir anahtar olabilir. Anahtarın alabilecegi olası değerler genişliğine anahtar uzayi denir.

Günümüzde kullanılmakta olan modern ve güçlü şifreleme algoritmaları artik gizli değildir. Bu algoritmalar güvenliklerini kullandıkları farkli uzunluk ve yapılardakı anahtarlarla sağlarlar. Bütün modern algoritmalar şifrelemeyi ve şifre çözmeyi kontrol için anahtarları kullanır. Bir anahtar ile şifrelenmiş bilgi, kullanılan algoritmaya bağlı olarak, ilgili anahtar ile çözülebilir. Genel olarak anahtarın kullanımi şu şekildedir (şekil 7):



Şekil 7. Bir mesajin dinlenmesini önlemek için şifreleme kullanılması.

Kullanıcı bir mesaji (m) göndermeden önce bir anahtar (k1) kullanarak şifreler. Şifreli metin (c) yasadişi dinleyicilere açık olan bir kanaldan gönderilir. Mesaji okumak için alıcı bir anahtar (k2) kullanarak şifreyi çözer ve m mesajini elde eder. Aktif düşmanlar araya girip iletişimi dinleyebilir. Eğer k1 ve k2 eşitse, sistem *simetriktir* . Aksi takdirde bu sistem *asımetrik* olarak nitelenir. Güvenliğin garantilenmesi için k2 her

zaman gizli olmalidir, ancak k1'i kullanarak k2'yi elde etmek mümkün olmadiği sürece k1 açıklanabilir. Bu durumda sisteme açık anahtarlı sistem (public key system) adi verilir.

1.5. Şifreleme Tekniklerinin Sınıflandırılması

Şifreleme teknikleri algoritmalarına, anahtar sayısına ve şifrelenecek mesajın tipine göre sınıflandırılmiştır. 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 kullanılmiştır.

Şifrelemenin güvenilirliği, algoritmanın kendisi saklı kaldığı müddetçe geçerlidir.

Büyük işletmeler için uygun bir şifreleme yöntemi değildir. Kullanılan algoritmayı bilen birinin isten ayrılması veya kazara algoritmanın açıga çıkması durumunda sistemin yeni bir algoritmaya göre yeniden güvenliğinin sağlanması gereklidir. Bu da işletmenin büyüklügüyle orantılı olarak artan, baslı basına hacimli bir istir. 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 kuruluslar 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[3-5].

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 çogunlugu açık algoritmaya dayalı teknikler kullanır. Algoritması açık olan şifreleme tekniğinde, bir veri şifrelenecegi zaman, şifreleme anahtarı kullanılır. Şifre çözüleceği zaman ise karşılık gelen şifre çözücü anahtar kullanılır.

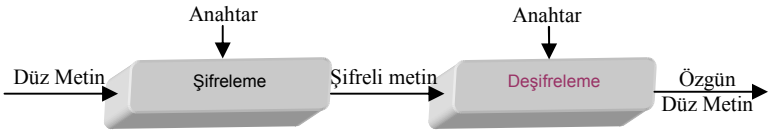
Bu anahtarın gizli tutulması son derece kritiktir çünkü bu anahtarı eline geçiren herhangi biri bütün mesajları çözebilecektir. Esasen şifreleme ve şifre çözme işlemleri oldukça kolaydır. Zor olan ise anahtarların güvenli bir şekilde saklanıp, gerekli olduğunda yine güvenli bir şekilde ilgili sahıslara gönderilmesidir.

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

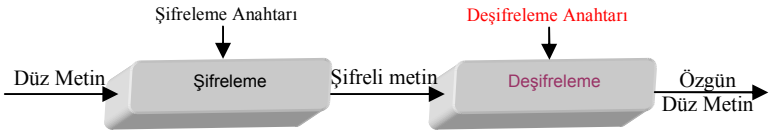
1.5.3. Tek ve Çift Anahtarlı Şifreleme Teknikleri

Açık anahtarlı sistemler pek çok ilginç olanaklar sunar; örneğin herkes online bir mağazaya mağazanın açık k1 anahtarını kullanarak şifrelenmiş bir kredi karti numarası gönderebilir. k2 anahtarını sadece maşaza bildiği için, kartin numarasını sadece mağaza öğrenebilir. Eğer simetrik sistem kullanılsaydi, mağaza potansiyel müşterilerinin her biriyle önceden ve gizlice ayrı ayrı anahtarlar belirlemek zorunda kalirdi. Açık anahtarlı sistemlerin güvenliği her zaman belirli matematiksel problemleri çözmenin zorluğuna dayanir, simetrik sistemler daha çok tek kullanımlik, geçici yapıdadırlar. Açık anahtarlı sistemlerin en büyük dezavantaji matematiksel yapıları nedeniyle simetrik sistemlerden

daha yavaş olmalarıdır; özellikle açık anahtarlı sistemlerdeki anahtarların boyutları simetrik sistemlerin anahtarlarının boyutlarından çok daha büyüktür. Kisaca, kullanılacak şifreleme yöntemi gerçekleştirilecek uygulamaya bağlı olarak seçilir. (Şekil 8.a, Şekil 8.b)



Şekil 8.a. Tek anahtar ile şifreleme ve şifre çözme



Şekil 8.b. Iki farkli anahtar ile şifreleme ve şifre çözme

Algoritmalardakı bütün güvenlik anahtara (veya anahtarlara) dayalıdir, hiçbiri algoritmanın ayrıntılarında yer almaz. Bu, algoritmanın yayınlanabildiği ve incelenebildiği anlamına gelir. Bu algoritmayı kullanan ürünler seri üretilebilir. Bir davetsiz misafirin sizin algoritmanizi bilmesi önemli değildir; sizin özel anahtarınızı bilmedikçe, o şahıs iletilerinizi okuyamaz. Bir şifre sistemi algoritmalardan ve olası bütün düz metinlerden, şifreli metinlerden ve anahtarlardan oluşur.

1.5.4. Gizli ve Açık Anahtar Yöntemleri

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

- Gizli-Anahtar (Simetrik) yöntemleri (Geleneksel şifreleme sistemleri)
- Açık-Anahtar (Asımetrik) yöntemleri (Açık anahtar şifreleme sistemleri)

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

1.5.4.1. Gizli Anahtar (Simetrik) Yöntemleri

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 anlasmaları ve bu anahtarı gizli tutmaları gerekmektedir. Eğer bu kişiler ayrı konumlarda bulunuyorsa, tasıyıcının, telefon sisteminin ya da Diğer tasıma 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 ugrasmak durumundadır. Anahtarların gizli kalmasını gerektirdiğinden dolayı, simetrik şifreleme, özel anahtar yönetiminde oldukça sıkıntı yaşamaktadır.

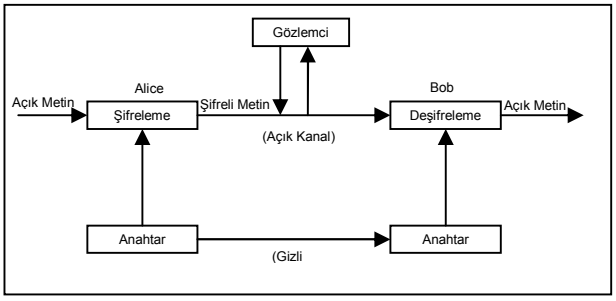
DES, Blowfish, Twofish, AES, CAST128, RC5 simetrik şifreleme algoritmalardır. Bu algoritmaların en büyük avantajı basit ve kolay uygulanabilir olmasıdır. Bununla birlikte daha hızlı ve verimlidirler. Ancak, şifreleme ve şifre çözme için aynı anahtarın kullanılıyor olması dezavantaj dogurur. Tek bir anahtarın güvenliğinin sağlanması zordur. Diğer sahıslara bu anahtarın güvenli olarak gönderilmesi sorununun yanı sıra, bu sahısların anahtarı ne kadar gizli tutacagı sorun teskil 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 dagıtılacagıdır. DES, şifrelenecek blogun iki parçaya bölünmesi ve her aşamada sadece biri üzerinde işlem yapılması esasına dayanır.

1.5.4.1.1. 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 anlastıkları anahtarı bilmiyorsa, şifreyi çözemez.



Şekil 9. Simetrik şifreleme

Şekil 9'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 seklindeki bir alfabede harflerin dizilisi, her bir harfin 3 harf saga kaydırılması sonucu DEFGHIJKLMNOPQRSTUVWXYZABC sekline dönüşmektedir. Bu şekilde, A harfinin şifreli karşılığı D harfi, B harfininki E harfi… seklinde olmaktadır. Günümüz standartları ile kıyaslandığında oldukça zayıf olan bu algoritma, geleneksel şifrelemenin nasıl çalıştıgına ilişkin basit bir örnek teskil etmektedir.

Şifreleme işleminin matematiksel notasyonlarla ifadesi su 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 akıs şifreleyiciler

1.5.4.1.1.1. Blok Şifreleyiciler

Blok şifreleyici, açık metindeki sabit boyutlu veri blogunu, şifreli metindeki aynı uzunluklu başka bir veri bloguna dönüştürür. Bu dönüşüm, kullanıcıya baglı bir gizli anahtar yardımıyla yapılır. Deşifreleme ise, aynı gizli anahtarı kullanarak, algoritmanın tersinin uygulanması ile sağlanır. Blogun sabit boyutu, çogu blok şifreleyicide 64 bit uzunlugundadır. 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.

1.5.4.1.1.2. Akıs Şifreleyiciler

Akıs şifreleyici algoritmalar, blok şifreleyici algoritmalara göre çok hızlı üretilebilirler. Blok şifreleyiciler büyük veri blokları üzerinde işlem yaparken, akıs şifreleyiciler açık metindeki genelde bit olan küçük veri birimleri üzerinde işlem yaparlar. Blok şifreleyici kullanarak herhanği bir boyuttaki açık metnin şifrelenmesi sonucu, aynı anahtar kullanıldığı sürece, aynı şifreli metin üretilir. Akıs şifreleyici ile açık metin parçaları, işleme sokuldukları zamana baglı olarak değişik şekilde şifrelenirler. Akıs şifreleyiciler, bit dizisi seklinde akıs anahtarları üretirler ve bu anahtarlar açık metinle XOR işlemine sokularak şifreleme işlemi gerçekleştirilir. Akıs anahtarının üretilmesi, açık metne ve şifreli metne baglı olabileceği gibi, veriye ve bu verinin şifreleme tarzına da baglı olabilir.

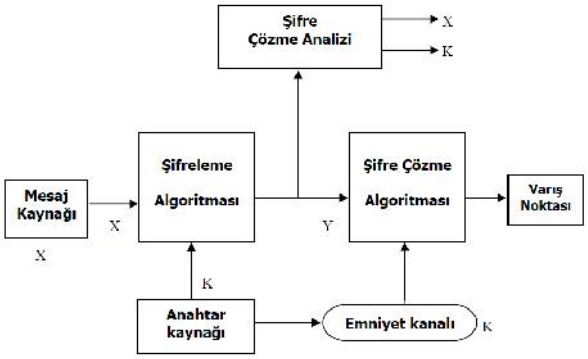
Açık metne veya şifreli metne baglı olarak akıs anahtarı üretilen sistemler, senkron akıs şifreleyici olarak, veriye ve şifreleme tarzına baglı olarak akıs anahtarı üretilen sistemler de, kendi kendine (*self*) senkron sistemler olarak adlandırılır.

1.5.4.1.2. Geleneksel Şifreleme Teknikleri

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

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

Geleneksel şifreleme güvenliği, çesitli faktörlere bağlıdir. Ilk olarak şifreleme algoritmasının çözülemeyecek kadar güçlü olması gerekir. Bunun ötesinde şifreleme güvenliği algoritmanın gizliliğine değil, anahtarın gizliliğine bağlıdir. Başka bir deyişle algoritma gizli tutmamali sadece anahtar gizli tutulmalidir. Geleneksel şifrelemenin çok yaygın kullanılmasını mümkün kilan bu özelliğidir. Algoritma gizliliğine gerek yoktur. Geleneksel şifrelemenin bu kullanımi güvenlik problemlerini anahtar gizliliği ile güvence altina alir.



Şekil 10. Geleneksel şifreleme sistem modeli

Şekil 10'da geleneksel şifreleme sistem modeli gösterilmiştir. Bir mesaj kaynağı olan X, anlaşilir mesaji üretir. Şifreleme için bir anahtar belirlenir. Orijinal mesaj için bir anahtar üretilmişse bu anahtar mesaj kaynağından bir güvenlik kaynağıyla karşı tarafa ulaştırilmalidir.X mesaji ve K şifreleme anahtarı şifreleme algoritmasına girdi olarak alinir.Bunun sonucunda şifreleme algoritması (E)nin ürettiği şifreli metin ortaya çikar (Y).

$$Y = E_K(X)$$

Bu notasyon Y şifreli metini E şifreleme algoritmasının bir fonksiyonu olduğunu gösterir. E şifreleme algoritması da K anahtarını kullanarak çalışir. Art niyeti olan bir kişi anahtara sahip olarak gerekli dönüşümü yapabilir.

$$X = D_K(Y)$$

Y şifreli metinini gözlemleyen ancak K şifreleme anahtarı veya X mesajina ulaşamayan bir kişi X veya K' yi yada her ikisini de ele geçirmeye çalışir.

Farz edelim ki bu art niyetli kişi şifreleme algoritmasını (Encryption-E) ve deşifre algoritmasını (Decryption -D) bilsin. Bu kişinin ilgisi sadece orijinal mesaj ise tahmini bir X mesaji oluşturarak X i elde etmeye çalışacak ve tahmini bir K şifreleme anahtarı oluşturup K anahtarını da elde etmeye teşebbüs edecektir.

1.5.4.1.3. Modern Şifreleme Teknikleri

Modern şifreleme teknikleri blok s eklinde şifreleme ve deşifreleme yapmaktadır. Hemen hemen bütün simetrik blok şifreleme algoritmaları " *Feistel blok şifresi* " olarak bilinen bir yapı üzerine kurulmuştur. Bu sebepten dolayı öncelikle bit-bit ve blok-blok şifreleme yapan algoritmalar hakkinda bilgi vermek konunun anlaşilması için faydalı olacaktır. Bu tekniklerden örnek olarak aşağida gelmişti:

- Veri şifreleme standardi (DES)
- Uluslar arası veri şifreleme algoritması (IDEA)
- Blowfish

1.5.4.2. Açık Anahtar (Asımetrik) Yöntemleri

Açık anahtarlı şifreleme, simetrik şifreleme algoritmalarından radikal bir farklilik göstermektedir. Bu şifreleme yöntemi iki ayrı anahtar kullanan yöntemdir. Iki anahtar kullanımının; güvenirlik, anahtar dağıtimi ve onaylama alanlarında önemli sonuçları vardir. Bu kriptografi yapısinda, açık ve gizli anahtar olarak adlandırılmış olan bir anahtar çifti kullanılmaktadır.

Asımetrik algoritmalar da denilen açık anahtarlı algoritmalarda şifreleme için kullanılan anahtar ile şifre çözme için kullanılan anahtar birbirinden farklidir. Anahtar çiftlerini üreten algoritmaların matematiksel özelliklerinden dolayı açık-gizli anahtar çiftleri her kişi için farklidir, diğer bir deyişle her kullanıcının açık-gizli anahtar çifti yalnızca o kullanıcıya özeldir. Ayrıca şifre çözüm anahtarı, şifre anahtarından hesaplanamaz (en azından makul bir zaman dilimi içerisinde).

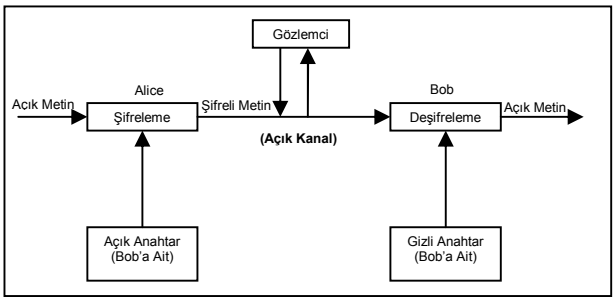
Bu algoritmalara açık anahtarlı algoritmalar adi verilmesinin sebebi şifre anahtarı halka (kamuya/genel kullanıma) açılabilir olmasıdir. Bir yabanci bir iletiyi şifrelemek için şifreleme anahtarını kullanabilir, ancak sadece ilgili şifre çözüm anahtarına sahip bir kişi iletinin şifresini çözebilir. Bu sistemde, şifre anahtarına genellikle açık anahtar denir, şifre çözüm anahtarı da genellikle gizli anahtar olarak adlandırılir. Gizli anahtar kimi zaman özel anahtar olarak da adlandırılir, ancak simetrik algoritmalarla karısmaması için bu terim genelde kullanılmaz.

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 dogurur. RSA, El Gamal, Knapsack, RC2, IDEA açık anahtarlı şifreleme tekniklerinden bazılarıdır.

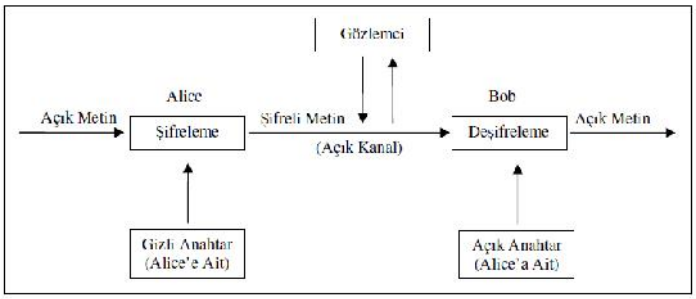
1.5.4.2.1. Açık Anahtar Kullanarak Şifreleme

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 11. Açık anahtarlı şifreleme-Senaryo 1

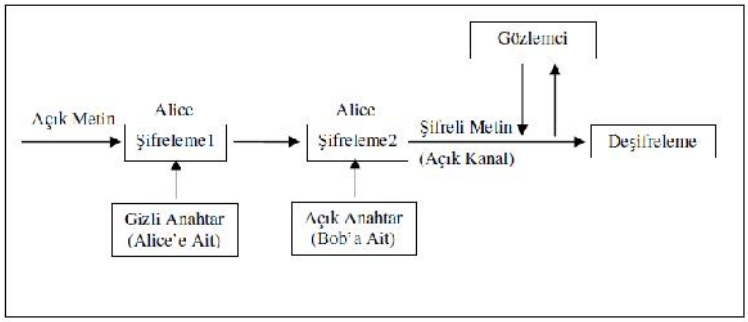
Bu senaryo ile Bob, sadece kendisinin okuduğundan ve başka herhangi bir kimsenin görüntüleyemediğinden emin olduğu bir mesaj alır. Fakat bunun kimden geldiğinden emin olamaz. Sadece gizlilik sağlanmiş olur. Senaryo su şekilde değiştirilmiş olsun:Eğer Alice, Bob'a, Bob'un Alice'den geldiğine emin olarak 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 ulasana 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.



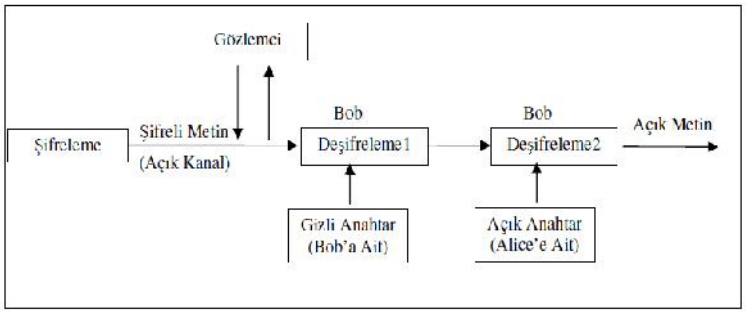
Şekil 12. Açık anahtarlı şifreleme-Senaryo 2

Bu senaryo ile de, gizlilik yerine kimlik denetimi sağlanmiş olur. Hem gizliliğin hem de kimlik denetiminin sağlanabileceği bir senaryo su 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 13. Açık anahtarlı şifreleme-Senaryo 3 (Şifreleme Kısmı)



Şekil 14. Açık anahtarlı şifreleme-Senaryo 3 (Deşifreleme Kısmı)

1.5.4.2.2. 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 yaygınlasan 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 oluşturulmus sayılar serisidir. Yani belgenin içeriğinin şifrelenerek saklanmasıdır. Sayısal imzalar göndericinin kimliğinin kesin bir biçimde teyit edilmesini ve elektronik dokümanın bütünlügünün kontrolünü mümkün kılar. İnkar edilemez özelliktedir (*Non-repudiation*).

Sayısal bir imza, kişinin el yazısı ile attığı imzaya es 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.

1.5.4.2.2.1. 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 akısı sırasında bilgilerin içeriğini korur, bir başka kişinin eline geçmesini ya da değiştirilmesini engeller, bilginin sadece alıcıya gittiğini ve sadece alıcı tarafından okunacagını garanti eder.

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

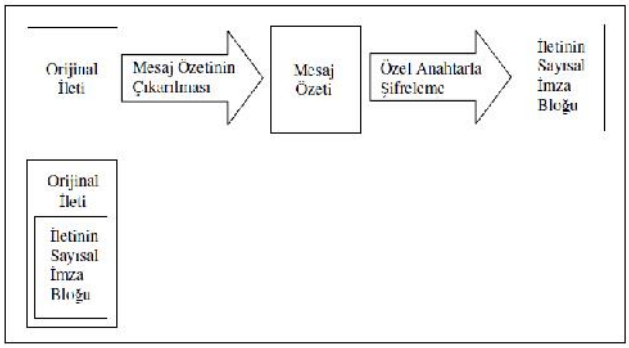
1.5.4.2.2.2. İletinin İmzalanması

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 tasarlanmiş bir MD (Message Diggest) algoritmasıdır. Bu algoritma, sonsuz uzunlukta veriyi girdi olarak kabul eder ve sonuçta 128 bit uzunlugunda bir çıktı üretir. SHA (Secure Hash Algorithm) ise, MD5'e benzeyen bir algoritmadır. MD5'le aralarında bir takım farklılıkları vardır: MD5'te çıktının uzunlugu 128 bit iken, SHA'da 160 bittir. Girdi olarak 264 – 1 uzunlugunda veriyi kabul eder. Ürettiği 160 bitlik sonuç ile kaba kuvvet ataklara karşı daha

dayanıklıdır. MD5 ve SHA'nın yanında bir çok MD algoritması tasarlanmiştır. 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- İetinin mesaj özetinin özel anahtarla şifrelenmiş hali iletinin sayısal imza blogudur ve bu değer iletinin sonuna eklenir, böylece ileti sayısal olarak imzalanmiş olur.



Şekil 15. İetinin sayısal imzalanması

1.5.4.2.2.3. İletinin İmzasının Doğrulanması

Gelen iletinin sayısal imzasının doğrulugunun kontrolü için, iletiyi imzalayan sahsın açık anahtarına ihtiyaç duyulmaktadır. Açık anahtarlar Sertifika Otoritesi tarafından halka açık yerlerde yayınlanmaktadırlar (Active Directory, LDAP ...). Doğrulama işlemi için gereken adımlar sunlardır :

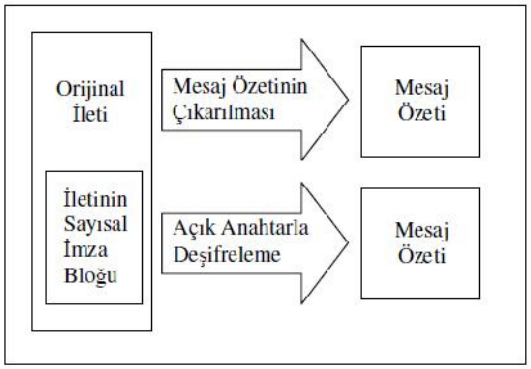
1- Orijinal ileti, iletinin sayısal imza blogu haricindeki kısmıdır. Bu kısmın mesaj özeti çıkarılır. Bu işlem için kullanılan algoritma imzalarken kullanılan algoritmayla aynı olmalıdır.

2- İletinin sonuna eklenen imza blogu 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 esit olması iletinin bozulmamiş 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 bozulmus olduğunu göstermektedir [7].



Şekil 16. Sayısal imzanın doğrulanması

1.5.4.2.2.4. Sayısal İmzanın İnkar Edilemez Özelliği

Sayısal imzanın inkar edilemez özellikte olmasını, imza doğrulama işleminin 2. adımında sayısal imza blogunun şifresini çözmek için kullanılan açık anahtar sağlamaktadır. Bu adımda özel anahtarla şifrelenen veri, açık anahtarla deşifre edilmektedir. Sayısal imzalama işleminde kullanılan asimetrik algoritmanın dogası geregi 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 blogunu basarıyla deşifre etmişse, imza blogunu imzalayan kişinin, açık anahtarın sahibi kişi olduğu kanıtlanmiş olur. Aksi halde, imza blogunun açık anahtar tarafından deşifre edilememesi durumunda, imza blogu bozulmustur (değiştirilmiştir) veya ileti açık anahtarın sahibi kişi tarafından imzalanmamiştır.

1.5.5. Şifreleme Sistemlerin İnceleme Boyutları

Şifreleme sistemleri genel olarak birbirinden bağımsiz üç boyutta incelenir:

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

2. ullanılan anahtarların sayısi: Bütün gönderici ve alıcılar ayni anahtarı kullanırsa sistem; simetrik, tek anahtarlı, gizli anahtarlı veya geleneksel şifreleme olarak bilinir. Eğer gönderici ve alıcı farkli anahtarlar kullanırsa bu asımetrik; iki anahtarlı yada kişisel anahtarlamali şifreleme olarak bilinir.

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

1.6. Kriptanaliz

X veya k, veya her ikisini bulması girişilmesi sürecine kriptanaliz denir. Tüm olası anahtarları denemeye kaba kuvvet (brute-force) yaklaşma denir .

Türkçe şifre çözüm diyebileceğimiz kriptanaliz, şifre bilimin (kriptolojinin) temel etkenlik alanıdır. Şifre yazımda, şifre çözümde 1940-1944 Dünya Paylaşım Savaşında oldukça gelişmiştir. Temel olarak, şifre çözümün amacı, kullanılan şifrenin zayıflıklarından ve şifrelenen metin hakkındaki bilgilerden yola çıkarak bütün anahtarları deneme zahmetinden kurtulmaktır. Kriptanalist, veya şifre çözücü düşmanın şifrelerini çözmekle uğraşan teknisyendir. Kriptolog ise şifre bilimcidir.

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

Alan Turing, Enigma şifresinin yapısal zayıflıklarını kullanarak, Colossus isimli ilk tüplü bilgisayar yardımıyla da Nazilerin iletişimlerini çözmeyi başardı (Colossus'ün bugünün adi dört işlemli hesap makinelerinden bile kat kat daha yavaş olduğunu göz ardı etmemek gerek). Şifrelemenin bütün esprisi düz metini (veya anahtarı, ya da her ikisini) davetsiz misafirlerden (istenmeyen kişilerden, düşmanlardan, saldırganlardan, rakiplerden) gizli tutmaktır. Şifre analizi anahtarı kullanmadan bir iletinin düz metnini eski haline getirme bilimidir. Bu işlem orijinal metin veya şifreleme anahtarı veya her ikisini de bulmaya yönelik bir işlemdir. Burada kullanılan strateji doğal olarak şifrelemenin tasarımına ve şifre çözen analistin bilgisine bağlıdır. Analistin sahip olduğu bilgi ve teçhizat olası saldırıları belirler. Başarılı bir şifre analizi düz metini veya anahtarı eski, ilk haline getirir.

Bir kriptografik tasarımın güvenliğini etkileyen faktörler makinelerin hesaplama gücü, anahtarın rasgeleliği, kullanılan anahtar ve algoritmanın kendisidir. Bir algoritma hesaplama kaynakları ile kırılamaz ise hesaplanabilir olarak güvenlidir. Ancak, hesaplama gücü son yıllarda çok gelişmiştir. Ayrıca çoğu algoritma saldırıları, paralel bilgisayarlarda gerçekleştirilmektedir. Algoritmanın performansını etkileyen diğer bir faktör de anahtar uzunluğudur. Saldırıları etkisiz kılmak için uzun anahtarlar kullanırız. Bir anahtar sıklıkla değiştirilmeli ve eksi anahtarlar yeniden kullanılmamalıdır.

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

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

Şifreli bir metin orijinal metini belirleyebilecek yeterli bilgi bulundurmuyorsa, bu şifreleme algoritması şartsız olarak güvenlidir. Burada kişinin ne kadar zamanı olduğu önemsizdir, şifreli metinin çözülmesi imkansızdır. Şifreleme algoritması seçilirken aşağıdaki kriterlerden biri yada her ikisine dikkat edilmelidir

- Şifreyi kırmanın maliyeti şifrelenmiş bilginin maliyetini aşmalı.

- Şifreyi kırmak için geçen zaman bilginin yaşam seyrindeki geçerlilik süresini aşmalı.

Eğer şifreleme yöntemi yukarıda sözü edilen iki kriteri karşılıyorsa bu yöntem hesaplama yönünden güvenlidir denir.En önemli saldırı tekniklerinden bazıları aşağıda verilmiştir.

1.7. Kaos ve Kriptoloji

1.7.1. 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.

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

- bir durumun değil bir sürecin bilimi,
- bir varoluşun değil bir oluşumun bilimi

- sistemlerin global doğasının bilimidir ve
- karmaşıklığın evrensel davranış biçimini sorgular (Gleick 1997; Faigenbaum 1981, Prigogine & Stengers 1984; Cramer, 1998).

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

Gökyüzünde hareket eden bulutlar, içilen bir sigaradan havaya yükselen duman bir musluktan akan su damlaları, bir rüzgarın etkisiyle oradan buraya salınıveren yapraklar... İlk bakışta her biri diğerinden farklı gibi görülen bu doğa manzaralarının içerdiği, bir ana tema vardır. Bütün bunlar ve yaşamımız boyunca her an tanık olduğumuz bir çok olay bize düzensiz, kuraldışı, devamsız olarak niteleyebileceğimiz oluşumlar olarak gözükür. Gerçekten de düzensiz ve her an öngörülemez bir geleceğe dönüşüveren doğa olayları, başta fizikçiler olmak üzere diğer doğa bilimcileri için bir sırdır. Bu özelliğinden dolayı, belirlenen tipte doğa durumlarına "bilinemezlik-karışıklık" anlamında "kaos" adı verilir. Kaos terimi, yalnızca adı geçen oluşumları tanımlamakla kalmaz, aynı zamanda bunları bilimsel çerçevede incelemenin, anlamanın ve kontrol etmenin de adı olur.

J.Gleick. kaos için şunları yazmıştır:

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

Kaosu ilgi çekici kılan ve bir devrim olarak değerlendirilmesine yol açan şey, kararlı olarak (belli sabit kanunlara göre) evrimleşen bir sistemin (hiç beklenmedik şekilde) düzensiz ve rast gele davranabilmesidir. Mesela bir adada yaşayan belli bir canlı

türünün sayısı, bu canlının üreme hızına, adadaki besin miktarına, bu canlıyla beslenen diğer türlerin adadaki etkinliği vs. gibi birtakım faktörlere belli oranda bağlıdır. Benzer şekilde bir benzin istasyonuna uğrayan arabaların veya âcil servise gelen hastaların geliş ve servis zamanlarının dağılışı bu çerçevede değerlendirilebilir.

Kolaylık olması için bütün bu faktörlerin zamanla değişmediği kabul edilerek yapılan en basit modellerde bile çok ilginç sonuçlar ortaya çıkmaktadır. Üreme hızını ifade eden bir katsayı belli bir değerden küçükse, çoğalma ve ölümleri ifade eden denklemler arasında bir denge kurulmakta ve nüfus belli bir değerde sabit kalmaktadır. Fakat bu katsayı, kritik bir değerden büyükse, nüfusun yıllara göre değişimi hiçbir formülle ifade edilemeyecek kadar düzensizleşmekte ve nüfus her yıl rast gele değerler alabilmektedir. Yani nüfus değişimi tamamen belirli kanunlara göre cereyan eden (deterministlik) bir hadise olmasına rağmen, sonuçta belirsizlik ve düzensizlik doğmaktadır. Benzeri bir durum, bir borudan akan sürtünmeli bir sıvının akış şekillerinde de görülmektedir. Akışkanların uyduğu dinamik denklemler Newton kanunlarından çıkartılıp düzgün akışlara gayet güzel bir şekilde uygulanabilmesine rağmen, akış hızı belli bir değeri aştığında girdaplar ve türbülans hareketleri oluşmakta ve akış tamamen kaotik hale gelmektedir.

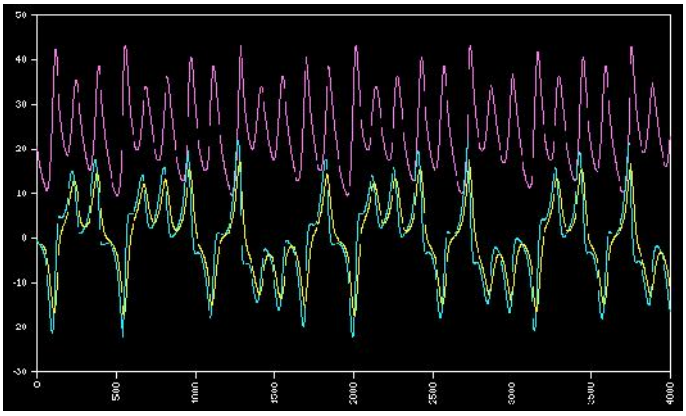
Kaos teorisi engin uygulama alanına sahip olan bir yaklaşımdır. Her türlü alanda uygulanabilme yeteneğinden dolayı, kaos teorisinin bilim dallarını birbirinden soyutlayan engelleri aştığı söylenebilir. Çok küçük görünen bir nedenin kendisinden çok daha büyük sonuçlara yol açabileceği mantığından hareket eden kaos kuramı, düzensizlik ve karmaşadan çok, bu düzensizlik içerisinde belli bir düzeni, düzenli düzensizliği anlamaya yöneliktir. Doğadaki şekillerin küp, koni, küre ve silindir gibi standart şekiller olmaktan çok, daha düzensiz görünümlü ve doğrusallıktan uzak şekiller olduğu noktasından hareket eden fraktal geometri kavramı da, kaos kuramı analizlerinde büyük faydalar sağlar.

Kaosun 1960’lı yıllarda Lorenz ile başladığı söylenebilir. Lorenz kaosun en önemli kavramlarından birini, başlangıç durumlarına hassas bağımlılığını bulmuştur. Ayrıca Lorenz’in çekicisi(Lorenz Attractor) de uzun yıllar kaosu tanımlamıştı. II. Dünya Savaşı sonunda. Massachusetis Teknoloji Enstitüsünde hava tahminleri üzerine bilgisayar destekli araştırmalarda bulunan E. Lorenz, 1960'ta icat ettiği minyatür meteoroloji

modeliyle meslektaşlarını şaşırtmıştır. Lorenz ilkel bilgisayarını kullanarak havayı en basit şekilde ifade edilebilen bir hale indirgemiştir.

Bilgisayarı, havadaki ısı-basınç ilişkilerini, rüzgarın yönünü, siklon gruplaşmalarını sayısal olarak, her gün 12 denklem yardımıyla kaydediyordu. Rüzgarlar ve hava sıcaklıkları, Lorenz' in yazıcısından satır satır dökülürken dünyadaki gerçekleşme biçimiyle aynı davranışı gösteriyordu. Dahası, bu listelenen değerlerden hareketle tahminlerde de "bulunabiliyordu. Biraz uğraşıdan sonra sayıları grafiğe de dökmeyi başardı.

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



Şekil 17. Yukarıdaki şekilde Lorenz'in hava tahminlerini yapmak için

kullandığı formüllerin sonucunda elde edilen iki ayrı grafik görülmektedir. Şekilde de görüldüğü gibi başlangıç değerlerinde yapılan küçük bir değişiklik sonuçta birbirinden farklı grafiklerin oluşmasına sebep olmaktadır. Bu olay, Lorenz'e en küçük detayların bile ileride çok büyük sonuçlara neden olabileceğini hiç bir zaman uzun süreli bir hava tahmini yapılamayacağını göstermişti. Sistem, ilk çıkış noktasıyla çok hassas olarak ilişkiliydi. Bu olayı J.H.Poincare şöyle ifade ediyordu: "Başlangıç şartlarındaki küçücük bir hata son olguda muazzam bir hataya neden olacaktır. Bu durumda, olacağı öngörmek olanaklı değildir."

Kaotik sistemlerde, sistemin zaman içindeki gelişimini tam olarak belirleyebilmek için başlangıç değerlerini sonsuz hassasiyetle bilmek gerekmektedir. Herhangi bir dinamik sistem için de bu geçerlidir, fakat doğrusal (lineer) sistemlerde hata zamanla doğrusal artmasına rağmen kaotik sistemlerde üsse bağlı olarak artmaktadır. Yani doğrusal bir sistemde, mesela bir gezegenin Güneş etrafındaki hareketinde, başlangıçtaki gözlemde yaptığımız hata 1 birim ise, zamanla 2,3,4,... şeklinde artacak; fakat kaotik bir sistemde, mesela atmosferde 10, 100, 1000, 10000,... şeklinde korkunç bir süratle artacaktır. Bunun sonucunda da uzun süreli hava tahminleri kesinlikle mümkün olmayacaktır.

Daha gelişmiş atmosfer modellerinin ve daha hızlı bilgisayarların kullanımıyla iyi tahminler yapılabileceğini düşünen bazı araştırmacılar, ABD'de *Control Data Cyber* 205 isimli bir süper bilgisayarı hava tahminleri için devreye soktular. Lorenz'in ilkel bilgisayarı saniyede 60 işlem yapabilirken, yeni super bilgisayarın hızı saniyede milyonlarca işlemle ölçülüyordu. Lorenz modelinde 12 denklemle yetinmesine rağmen Control Data Cyber'e 500.000 denklem yüklenmişti.

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

1.7.2. Kaotik Özellikler

1.7.2.1. İterasyon

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

$$y = x^2 + c \;\; ; \;\; x = y \tag{1}$$

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

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

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

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

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:

$$f^n \to \begin{cases} \infty & |x| > 1 \\ 1 & |x| = 1 \\ 0 & |x| < 1 \end{cases} \quad n \to \infty \tag{4}$$

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 1'da gösterilmektedir.

Tablo 1 $f : x \to x^2 + ¼$ 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 2'de, 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 [8].

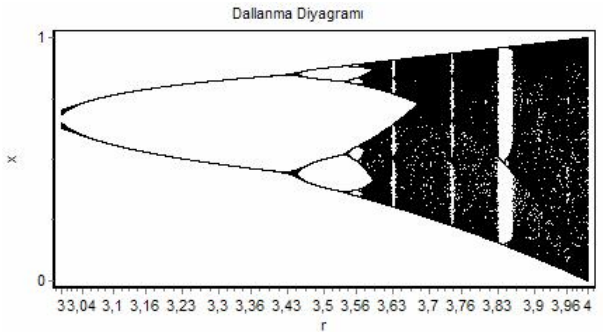
Ek olarak, başlangıçta birbirine oldukça yakın olan çekirdek değerleri, birkaç iterasyon sonrasında, oldukça farklı yörüngeler oluşturmaya başlar. Bu durum, kaosun temel karakteristiği olan, başlangıç koşullarına aşırı duyarlılık olarak adlandırılır. Davranış kaotik davranıştır ve bu durumun olmasına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

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

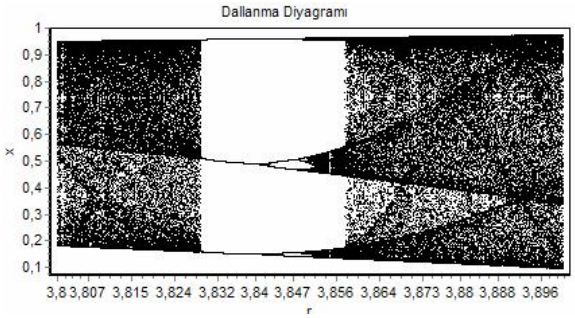
±1.75	**±1.5**	**±1**	**±0.75**	**±0.5**	**±0.25**
2.31	1.5	0.25	-0.1875	-0.5	-0.6875
4.59	1.5	-0.6875	-0.71484375	-0.5	-0.2773437
20.38	1.5	-0.2773437	-0.2389984	-0.5	-0.6730804
414.93	1.5	-0.6730804	-0.6928797	-0.5	-0.2969627
172173.29	1.5	-0.2969627	-0.2699176	-0.5	-0.6618131
2.964e+10	1.5	-0.6618131	-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
Tasma	1.5	-0.6449962	-0.6553090	-0.5	-0.3339798
⋮	⋮	⋮	⋮	⋮	⋮
sonsuz	1.5	-0.5	-0.5	-0.5	-0.5

1.7.2.2. Dallanma Diyagramı

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



Şekil 18. Lojistik haritaya ait dallanma diyagramı, r=[3,4]

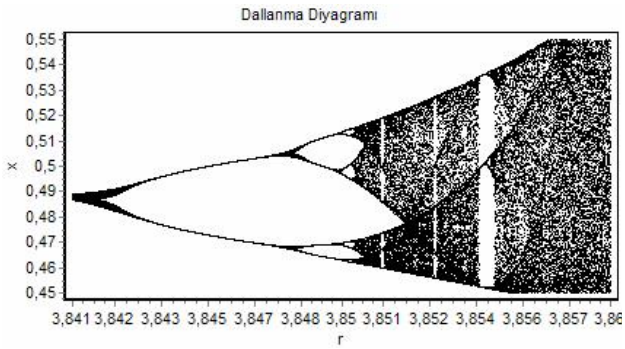


Şekil 19. Lojistik haritaya ait dallanma diyagramı, r=[3.8,3.9]

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.

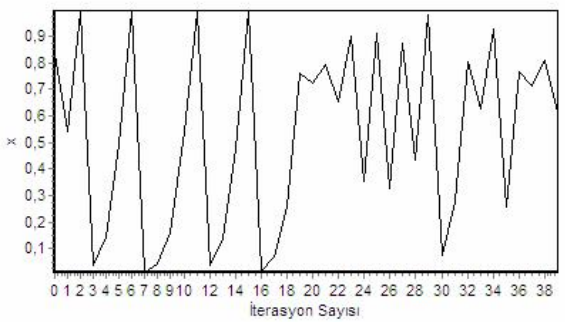
Şekil 18, 19 ve 20'de dallanma diyagramlarından bazı kısımlar gösterilmektedir. Bu şekiller incelenirse, dallanma diyagramlarının belirli bölgeleri alınıp büyütüldüğünde, ana şekle benzerlik gösterdigi görülür. Bu duruma kendine benzerlik denir ve fraktal yapılarda da gözlenir. Kaotik sistemler de fraktal yapıdadır. Ancak her fraktal kaotik özellik göstermez.



Şekil 20. Lojistik haritaya ait dallanma diyagramı, r=[3.84,3.86]

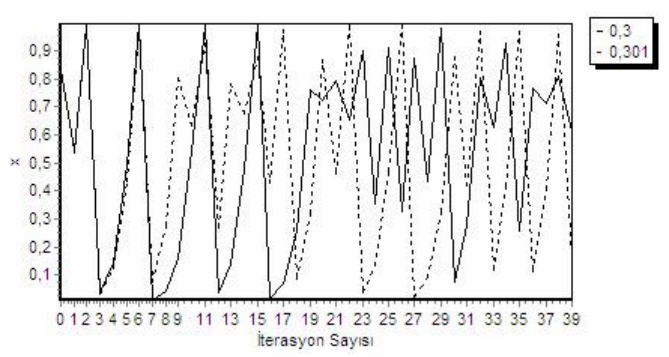
1.7.2.3. Başlangıç Koşullarına Duyarlılık

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 [9, 10]. Lojistik haritada r=3.99 ve x1=0.3 başlangıç değeri için zaman dizisi Şekil 21'de gösterildiği gibi olmaktadır.



Şekil 21. Lojistik haritada iterasyon sonucu oluşan değerler

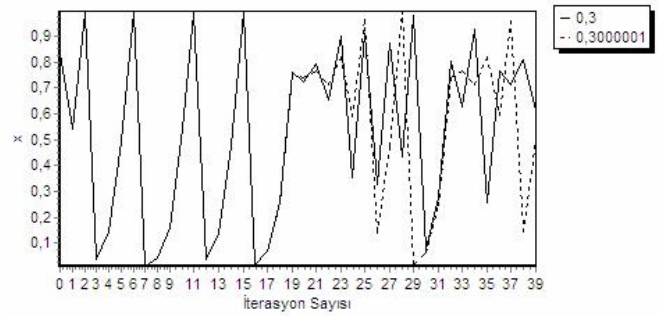
Başlangıç değeri olarak 0.301 seçildiğinde sistemin ürettiği değerler, başlangıç koşulu 0.3 olan sisteme göre farklılaşmaktadır. Bu durum, Şekil 22'deki grafikte gösterilmektedir.



Şekil 22. 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 23'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 23. Başlangıç değerleri arasında 0.000001'lik farkın sonucu

1.7.2.4. 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.

1.7.2.4.1. 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.

$$K = \{k_1, k_2, k_3, ..., k_r\} \tag{5}$$

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.

$$F(x) = \begin{cases} x/p, & x \in [0, p] \\ (1-x)/(1-p), & x \in [p, 1] \end{cases} \tag{6}$$

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 [11, 12].

1.7.2.4.2. 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 kosullarında 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.

1.7.2.4.2.1. 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 üstünlügü, 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 [13].

1.7.2.4.2.2. 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 eşlestirmeleri 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 seçildiginde 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 islem gücü gerektirmektedir. Çünkü bu işlemi yapan geleneksel algoritmalar için gerekli fonksiyonların temel işlemleri (aritmetik ve lojik kapılar), zaten mikroişlemcide mevcuttur [13, 14, 15].

1.8. DNA Hesaplama

1.8.1. DNA'nin Tarihi

1940'larda seçkin bilim insanı Barbara McClintock bir mısırın DNA'sının bir kısmına zarar verdi. Daha sonra şaşkınlık içerisinde bitkinin hasar görmüş kısmını tamir edebildiğini gözlemledi. Bunu DNA dizilimdeki diğer kısımlarını kopyalayıp, hasar gören yere yapıştırarak yapıyordu. Keşfi o zamanlar için oldukça radikal bir şeydi, nerdeyse raporlarına kimse inanmamıştı. (40 yıl sonra bu çalışma sebebiyle Nobel aldı.)

Ve bizler hala meraktayız. Bu denli ince bir hücre bunu yapmayı nasıl biliyor..?

Fransız bir HIV virüsü araştırmacısı ve bilgisayar bilimci, bu sorunun cevabının bir kısmını buldu. İpucu; DNA'daki yönergeler sadece dilbilimsel değildir, aynı zamanda harikulade bir matematiğe sahiptirler. DNA'nın yapısına hükmeden bir **Evrimsel Matris** vardır. Bilgisayarlar veri hatalarını belirlemek için '' sağlama toplamı '' adı verilen bir şey kullanırlar. Görünüşe bakılırsa DNA 'da bu tip bir sağlama (checksum) kullanıyor gözükmektedir. Fakat DNA'nın sağlaması sadece kayıp veriyi belirlemekle kalmaz; bazen de neyin kayıp olduğunu hesap edebilmektedir. Şimdi bunun nasıl çalıştığına bakalım;

İngilizcede E harfi metinler içerisinde 12.7 % yüzde ile görülmektedir. Z harfinin görülme sıklığı ise sadece 0.7% dir. Diğer harflerin görülme sıklığı da bu ikisinin arasında bir yerlere tekabül eder. Böylece İngilizcede sadece harfleri sayarak hata belirlemesi yapabilmek mümkündür.

DNA'da, bazı harfler diğerlerine göre tıpkı İngilizcedeki E gibi daha sık görülmekte, bazıları da daha az sıklıkta görülmektedir. Fakat..İngilizcenin tersine , bu harflerin hangi sıklıklarda görüleceği genetik koda saklanmış kesin bir matematiksel formülle kontrol edilmektedir.

Hücreler çoğaldığında, DNA dizimindeki tüm harflerin toplam sayısı hesap edilir. Harflerin toplam sayımı kesin bir oranla eşleşmezse, hücre bir hata yapıldığını bilir. Böylece işlemi durdurarak yeni hücreyi öldürür. Bu sağlama mekanizmasının işlevsizliği yapısal doğum bozuklukları ve kansere neden olur.

1.8.2. DNA'nin İlkeleri

DNA'nın dört ana bazı vardır. Sembolleri şu şekildedir; T (Thymine) ,C (Cytosine) , A (Adenine) ve G (Guanine). Bu semboller üçlü küme kombinasyonlar şeklinde gruplanırlar. Bunlar 4*4*4 = 64 olası kombinasyona sahiptir. Böylece genetik alfabe 64 harflidir. 64 harf amino asitleri ve proteinleri meydana getiren yönergeleri yazmakta kullanılır. Perez harfleri sayarken, eğer DNA'daki harfleri T-C-A-G tablosu şeklinde düzenlendiğinde ilginç bir örüntü göründüğünü ortaya koydu. Tabloyu gördüğünüz gibi ortadan 2'ye böldü (Tablo 3). İnsan genomunda 1 milyon üçlü kümeye sahip bir DNA dizimini aldı. DNA'daki her üçlü kümenin popülasyonunu saydı ve her bir aralığa yerleştirdi:

Tablo 3. Perezin düzenlediği ilk tablo

First Letter	Secound Letter			
	T	C	A	G
T	TTT	TCT	TAT	TGT
	TTC	TCC	TAC	TGC
	TTA	TCA	TAA	TGA
	TTG	TCG	TAG	TGG
C	CTT	CCT	CAT	CGT
	CTC	CCC	CAC	CGC
	CTA	CCA	CAA	CGA
	CTG	CCG	CAG	CGG
A	ATT	ACT	AAT	AGT
	ATC	ACC	AAC	AGC
	ATA	ACA	AAA	AGA
	ATG	ACG	AAG	AGG
G	GTT	GCT	GAT	GGT
	GTC	GCC	GAC	GGC
	GTA	GCA	GAA	GGA
	GTG	GCG	GAG	GGG

Harfleri topladığında, beyaz harflerin toplamı ile siyah harflerin toplamının birbirine oranı 1:1 di. Ve bu öyle kaba taslak olarak ortaya konmamıştı. Bu oran tam anlamıyla binde birlik bir orandan daha bile iyiydi. 1.000:1.000 . Daha sonra Perez tabloyu bu şekilde böldü (Tablo 4) :

Tablo 4. Perezin DNA harfları ile düzenlediği ikinci tablo

First Letter	Secound Letter			
	T	C	A	G
T	TTT	TCT	TAT	TGT
	TTC	TCC	TAC	TGC
	TTA	TCA	TAA	TGA
	TTG	TCG	TAG	TGG
C	CTT	CCT	CAT	CGT
	CTC	CCC	CAC	CGC
	CTA	CCA	CAA	CGA
	CTG	CCG	CAG	CGG
A	ATT	ACT	AAT	AGT
	ATC	ACC	AAC	AGC
	ATA	ACA	AAA	AGA
	ATG	ACG	AAG	AGG
G	GTT	GCT	GAT	GGT
	GTC	GCC	GAC	GGC
	GTA	GCA	GAA	GGA
	GTG	GCG	GAG	GGG

Perez'in keşfettiği beyaz harflerin siyah harflere oranının tam olarak 0.690983, (3 pi) /2. Pi ise 1.618 yani altın oranın sayısıdır. Daha sonra keşfettiği başka bir şey ise aynı oranın, yani 0.690983 tabloyu şu iki şekilde böldüğünde bile değişmemesiydi (Tablo 5):

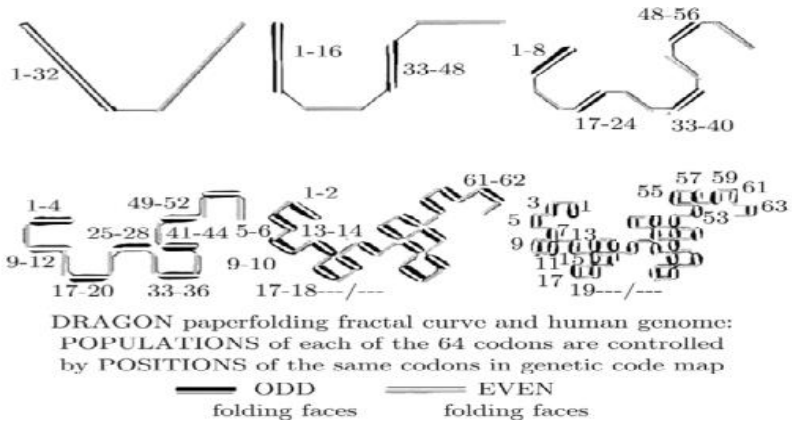
Tablo 5. Perezin DNA harfları ile düzenlediği üçüncü tablo

First Letter	Secound Letter			
	T	C	A	G
T	TTT	TCT	TAT	TGT
	TTC	TCC	TAC	TGC
	TTA	TCA	TAA	TGA
	TTG	TCG	TAG	TGG
C	CTT	CCT	CAT	CGT
	CTC	CCC	CAC	CGC
	CTA	CCA	CAA	CGA
	CTG	CCG	CAG	CGG
A	ATT	ACT	AAT	AGT
	ATC	ACC	AAC	AGC
	ATA	ACA	AAA	AGA
	ATG	ACG	AAG	AGG
G	GTT	GCT	GAT	GGT
	GTC	GCC	GAC	GGC
	GTA	GCA	GAA	GGA
	GTG	GCG	GAG	GGG

Yine, beyaz harflerin toplam sayısının siyah harflerin toplam sayısına oranı 0.6909 yani 1,000 de birlik bir orandan daha iyi bir kesinlikteydi. Perez üç tane daha farklı simetri keşfetti Ve şimdiye buraya kadar ki 3 yolla tablonun bölünmesi ile beyazların siyahlara oranı 1.000: 1.000 Ve diğer 3 yolla bölünmesine oranı 0.600983 ya da (3pi)/2

Bu 6 simetriyi birbirlerinin üzerine gelecek şekilde üst üste yerleştirirseniz, 32 altın basamaktan oluşan bir matematiksel merdiven elde edersiniz. Daha sonra da olağanüstü bir geometrik örüntü ortaya çıkar: Fraktal geometride çok iyi bilinen

‘‘ Ejderhanın Eğrisi ’’ karşınızdadır. DNA harfleri ile sınıflandırılmış giderek azalan frekanstaki sahip olan bir örneği (şekil 24) :



Şekil 24. DNA harfleri ile sınıflandırılmış giderek azalan frekanstaki sahip olan örnek

1.8.3. Diğer İlginç Gerçekler

Bu kuralların varyasyonlarına sahip benzer örüntüler, AIDS virüsünden bakteriye, primatlardan (insanımsılardan) insanlara kadar 20 farklı tür içerisinde gözlenmektedir.

DNA’daki her karakter belirlenmiş kesin sayılarda ortaya çıkar ve her birinin ikizi vardır. TTT ve AAA ikizdirler ve en sık rastlanılanlardır; İngilizcedeki E harfinin DNA’daki karşılıklarıdır.

Bu örüntü her biri için kendine özgü bir frekansa sahip 32 frekans oluşturur.

T ile başlayan üçlü kümelerin sayısı ile A ile başlayan üçlü kümelerle sayısı tam olarak aynıdır. (0.1% içerisinde)

C ile başlayan üçlü küme sayısı ile G ile başlayan üçlü küme sayısı tam olarak aynıdır.

Genetik kod tablosu fraktaldır. Yani aynı örüntü kendisini her seviyede tekrar eder. Mikro ölçüm kontrolü üçlü kümelerin amino asitlere dönüşmesini kontrol eder. Bu her biyoloji kitabında yazar. Perez tarafından keşfedilen makro ölçüm ise tüm organizmanın yapısal bütünlüğünü denetler.

Perez örüntüler içerisinde başka örüntülerde keşfetmiştir.

Burada yaptığım sadece aysbergin ucunu göstermekten ibaret. Daha burada makaleyi anlaşılabilir tutmak için çıkardığım birçok kural ve katman detayı

bulunmaktadır. Perez araştırmalarına devam etmektedir. Eğer Fransızca biliyorsanız, kitabı ‘‘ Codex Biogenesis ’’ veFransızca web sitesini tavsiye ediyorum. Burada İngilizce Çevirisi bulunmaktadır.

Yeri gelmişken, daha önce ‘‘ Çöp DNA’’ olarak adlandırıla gelen bölgenin hiçte çöp olmadığını ortaya koyan oldukça ilginç veriler bulduğunu da belirtmeliyim.

Kopyalama hataları evrimsel ilerlemenin kaynağını oluşturamaz, çünkü bu doğru olsaydı, sonunda tüm harflerin eşit şekilde olası olması gerekirdi.

Bu yararlı evrimsel mutasyonların rastgele olmadığını kanıtlar. Bunun yerine, bunların oldukça kesin bir evrimsel matris tarafından kontrol edildiği gösterir.

Organizmalar yatay gen transferi ile birbirlerinden gen alış verişi yaptıklarında, nihai sonuç hala kendine özgü matematiksel bir örüntüye tabidir.

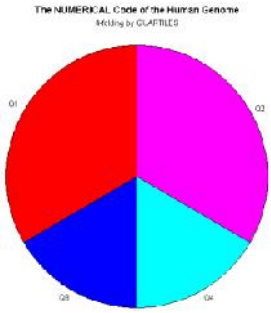
DNA yok edilmiş veriyi sağlama işlemi yoluyla eksilmiş içeriği yani romandan koparılmış sayfayı, tersine bir şekilde hesaplayarak, yeniden oluşturabilir.

Hiçbir insan yapısı dil böylesine kesin bir matematiksel yapıya sahip değildir. DNA oldukça sıkıca dokunmuş, oldukça özgün kuralları izleyen yüksek oranda etkin bir yapıdır. Alfabesi, grameri ve tüm yapısı harikulade matematiksel işlevlerle düzenlenmiştir.

1.8.4. Daha da İlginç Faktörler

En sık rastlanılan harf çifti (TTT ve AAA) insan genomu ve şempanzelerde tutarlı bir şekilde tam olarak 1/13X kez diğer harflerden daha sık gözükmektedir.

Grup 1’e en sık rastlanılan 32 üçlü kümeyi koyarsanız ve 32 en az sıklıkta rastlanılanları da Grup 2 ‘ye alırsanız, Grup 1 Grup ‘ ‘ye oranı tam olarak 2:1′dir . Ve üçlü kümeler simetrik çiftler halinde meydana geldikleri için (TTT-AAA, AAA-TTT vb.) , bunları 16 grup altında toplayabilirsiniz. Bu 4 adet üçlü kümeler popülasyonunu grafiksel olarak gösterdiğimizde, barış sembolünü elde ediyoruz:



Şekil 25. Üçlü kümeler popülasyonunu

DNA hem kendi kendini onaran, kendi kendini düzelten, kendini yazan ve kendi evrimini üstlenen bir koddur. DNA, insan mühendislerin ancak rüyalarında görebileceği, oldukça incelikli bir mühendislik seviyesi sergiler. Hepsinden ötesi, DNA zarif bir yapıdır.

Kanser bazen cinnetten çıkmış evrim diye tanımlanmıştır. Dr. Perez kanser hücrelerinde bu matriste oluşan bozunmaları da göstermektedir. Kuvvetle muhtemel olduğunu düşündüğüm şey, kanser araştırmalarında yeni atılımların bu matrisin içerisinde gizlendiğidir.

Tıp ve bilgisayar biliminde yürütülebilecek en üretken araştırmanın DNA evrimsel matrisi olduğunu ileri sürüyorum. Söylediğim gibi, bu sadece aysbergin uç kısmı.

Burada daha keşfedilecek çok şey var!

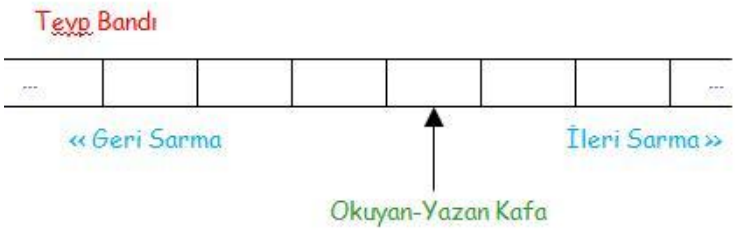
DNA dilini temel alan bilgisayar dilleri geliştirdiğimizde, bunlar olağan üstü veri sıkıştırma, hata düzeltme ve evet, kendi kendine evirilebilme yeteneğine sahip olabilecekler. Düşünün; bunlar kendi kendilerine yeni özellikler ekleyen ve zaman içinde bunu geliştiren bilgisayar programları olacak. Hem de tümünü kendileri gerçekleştirebilecek.

1.9. Turing Makinesi (Turing Machine)

Bilgisayar bilimlerinin önemli bir kısmını oluşturan otomatlar (Automata) ve Algoritma Analizi (Algorithm analysis) çalıştırmalarının altındaki dil bilimin en temel taşlarından birisidir.1936 yılında Alan Turing tarafından ortaya atılan makine tasarımı günümüzde pekçok teori ve standardın belirlenmesinde önemli rol oynar.

1.9.1. Turing Makinesinin Tanımı

Basitçe bir kafadan (head) ve bir de teyp bandından (tape) oluşan bir makinedir.(Şekil 26)



Şekil 26. Turing Makinasinin basit gösterimi

Makinede yapılabilecek işlemler

- Yazmak
- Okumak
- Bandı ileri sarmak
- Bandı geri sarmak

şeklinde sıralanabilir.

1.9.1.1. Chomsky Hiyerarşisi ve Turing Makinesi

Bütün teori bu basit dört işlem üzerine kurulmuştur ve sadece (1.9.1) deki bahs edilen işlemleri kullanarak bir işin yapılıp yapılamayacağı veya bir dilin bu basit 4 işleme indirgenip indirgenemeyeceğine göre diller ve işlemler tasnif edilmiştir.

Şekil 27. Turing Makinesinin kabul ettiği diller

Bu sınıflandırma Şekil 27 dakı venn şeması ile gösterilmiştir. Aynı zamanda chomsky hiyerarşisi (chomsky hierarchy) için 1. seviye (type-1) olan ve Turing makinesi ile kabul edilebilen diller bütün tip-2 ve tip-3 dilleri yani içerk bağımsız dilleri ve düzenli dilleri kapsamaktadır. Ayrıca ilave olarak içerik bağımsız dillerin işleyemediği (üretemediği veya parçalayamadığı (parse)) $a^n b^n c^n$ şeklindeki kelimeleri de işleyebilmektedir. Düzenli ifadelerin işleyememesi konusunda bilgi için düzenli ifadelerde pompalama savı (pumping lemma in regular expressions) ve içerik bağımsız dillerin işlemeyemesi için de içerik bağımsız dillerde pompalama savı (pumping lemma for CFG) başlıklı yazıları okuyabilirsiniz.

1.9.1.2. Turing Makinesinin Akademik Tanımı

Turing makineleri literatürde akademik olarak aşağıdaki şekilde tanımlanır:

$$M = (Q, \Sigma, \Gamma, \delta, q_0, \Diamond, F) \tag{6}$$

Denklem 6 da M ile gösterilen makinenin parçaları aşağıda listelenmiştir:

Q sembolü sonlu sayıdaki durumların kümesidir. Yani makinenin işleme sırasında aldığı durumardır.

Γ sembolü dilde bulunan bütün harfleri içeren alfabeyi gösterir. Örneğin ikilik tabandaki sayılar ile işlem yapılıyorsa {0,1} şeklinde kabul edilir.

Σ sembolü ile makineye verilecek girdiler (input) kümesi gösterilir. Girdi kümesi dildeki harfler dışında bir sembol taşıyamayacağı için $\Sigma \subseteq \Gamma$ demek doğru olur.

δ sembolü dilde bulunan ve makinenin çalışması sırasında kullanacağı geçişleri (transitions) tutmaktadır.

◊ sembolü teyp bandı üzerindeki boşlukları ifade etmektedir. Yani teyp üzerinde hiçbir bilgi yokken bu sembol okunur.

q_0 sembolü makinenin başlangıç durumunu (state) tutmaktadır ve dolayısıyla $q_0 \subseteq Q$ olmak zorundadır.

F sembolü makinenin bitiş durumunu (state) tutmaktadır ve yine $F \subseteq Q$ olmak zorundadır.

1.9.1.2.1. Örnek Turing Makinesi

Yukarıdaki sembolleri kullanarak örnek bir Turing makinesini aşağıdaki şekilde inşa edebiliriz. Örneğin basit bir kelime olan a düzenli ifadesini (regular expression) Turing makinesi ile gösterelim ve bize verilen aaa şeklindeki 3 a yı makinemizin kabul edip etmediğine bakalım. Tanım itibariyle makinemizi aşağıdaki şekilde tanımlayalım:

$$M = \{ \{q_0,q_1\} , \{ a \} , \{ a,x \} , \{ q_0\ a \rightarrow a\ R\ q_0 , q_0\ x \rightarrow x\ L\ q_1 \} , q_0 , x , q_1 \} \qquad (7)$$

Denklem 7 deki bu makineyi yorumlayacak olursak:

Q değeri olarak $\{q_0,q_1\}$ verilmiştir. Yani makinemizin ik idurumu olacaktır.

Γ değeri olarak { a,x } verilmiştir. Yani makinemizdeki kullanılan semboller a ve x'ten ibarettir.

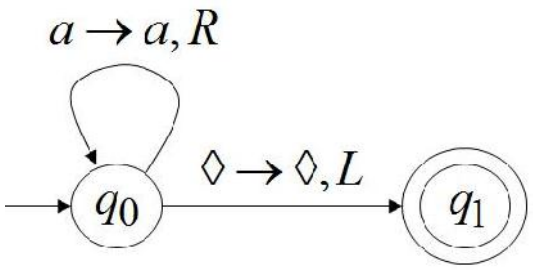
Σ değeri olara {a} verilmiştir. Yani makinemize sadece a girdisi kabul edilmektedir.

δ değeri olarak iki geçiş verilmiştir $\{ q_0\ a \rightarrow a\ R\ q_0 , q_0\ x \rightarrow x\ L\ q_1 \}$ buraadki R sağa sarma L ise sola sarmadır ve görüleceği üzere Q değerindeki durumlar arasındaki geçişleri tutmaktadır.

◊ değeri olarak x sembolü verilmiştir. Buradan x sembolünün aslında boş sembolü olduğu ve bantta hiçbir değer yokken okunan değer olduğu anlaşılmaktadır.

q_0 ile makinenin başlangıç durumundaki hali belirtilmiştir.

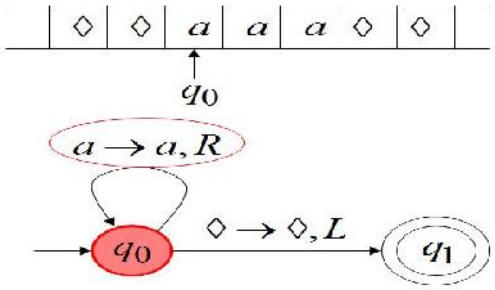
F değeri olarak q_1 değeri verilmiştir. Demek ki makinemiz q_1 durumuna geldiğinde bitmektedir (halt) ve bu duruma gelmesi halinde bu duruma kadar olan girdileri kabul etmiş olur. Yukarıdaki bu tanımı görsel olarak göstermek de mümkündür (Şekil 28) :



Şekil 28. Örnek Turing Makinesinin görsel olarak gösterimi

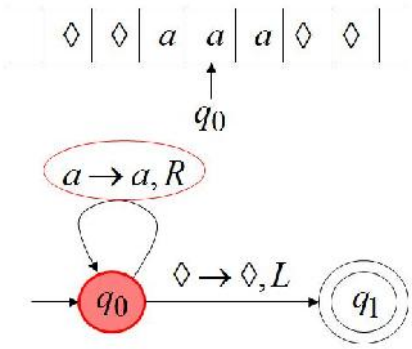
Yukarıdaki bu temsili resimde verilen turing makinesi çizilmiştir. Makinemizin örnek çalışmasını ve bant durumunu adım adım inceleyelim.

Birinci adımda bandımızda aaa (3 adet a) yazılı olduğunu kabul edelim ve makinemizin bu aaa değerini kabul edip etmeyeceğini adım adım görelim. Zaten istediğimiz de aaa değerini kabul eden bir makine yapabilmekti .



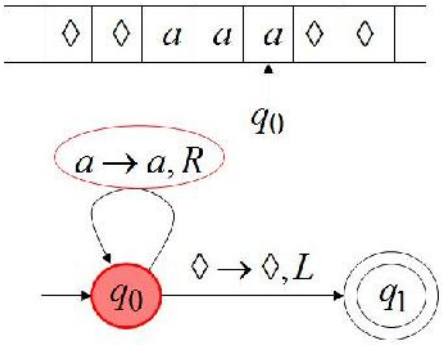
Şekil 29. Örnek Makinenin ilk adımı

Şekil 29 gibi, ilk durumda bant üzerinde beklenen ve kabul edilip edilmeyeceği merak edilen değerimiz bulunuyor. Makinemizin kafasının okuduğu değer a sembolü. Makinemizin geçiş tasarımına göre q_0 halinde başlıyoruz ve a geldiğinde teybi sağa sarıp yine q_0 durumunda kalmamız gerekiyor.



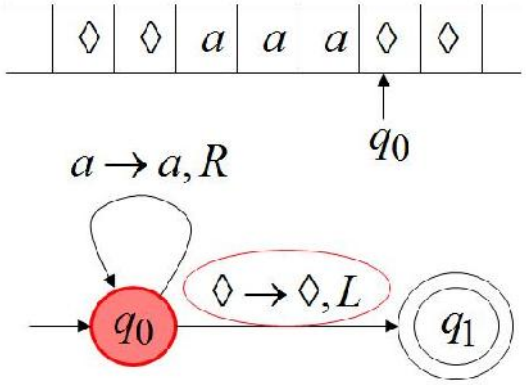
Şekil 30. Örnek Makinenin ikinci adımı

Yeni durumda (Şekil 30) kafamızın okuduğu değer banttaki 2. a harfi ve bu durumda yine q_0 durumundayken teybi sağa sarıp yine q_0 durumunda kalmamız tasarlanmış



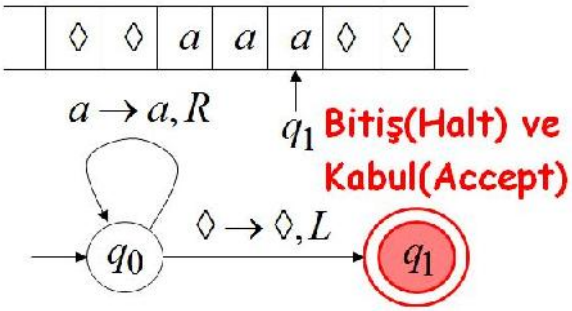
Şekil 31. Örnek Makinenin üçüncü adımı

3. durumda (Şekil 31) kafamızın okuduğu değer yine a sembolü olmakta ve daha önceki 2 duruma benzer şekilde q_0 durumundayken a sembolü okumanın sonucu olarak teybi sağa sarıp $q_0$durumunda sabit kalıyoruz.



Şekil 32. Örnek Makinenin dördüncü adımı

4. adımda (Şekil 32) teypten okuduğumuz değer boşluk sembolü x oluyor. Bu değer makinemizin tasarımında q_1 durumuna gitmemiz olarak tasarlanmış ve teybe sola sarma emri veriyoruz.

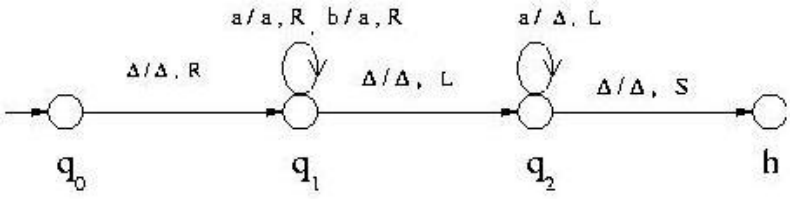


Şekil 33. Örnek Makinenin bitiş adımı (Kabul durum)

Makinenin son durumunda (Şekil 33) q_1 durumu makinenin kabul ve bitiş durumu olarak tasarlanmıştı (makinenin tasarımındaki F kümesi) dolayısıyla çalışmamız burada sonlanmış ve giriş olarak aaa girdisini kabul etmiş oluyoruz.

1.9.1.2.2. İkinci Örnek

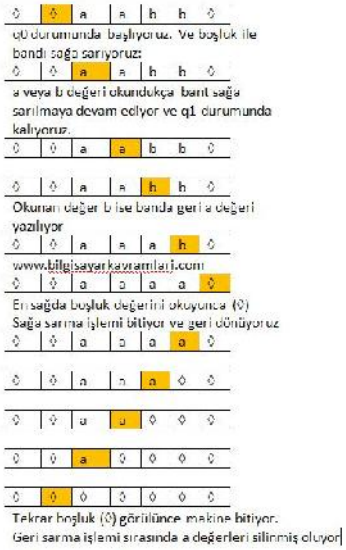
Makinemiz {a,b} sembolleri için çalışsın ve ilk durum olarak bandın en solunda başlayarak bantta bulunan sembolleri silmek için tasarlansın. Bu tasarımı aşağıdaki temsili resimde görülen otomat ile yapabiliriz:



Şekil 34. İkinci örneğin Turing Makinesi

Şekil 34 te görüldüğü üzere makinemizde 4 durum bulunuyor, bunlardan en sağda olan h durumu bitişi (halt) temsil ediyor. Şimdi bu makinenin bir misal olarak "aabb" yazılı bir bantta silme işlemini nasıl yaptığını adım adım izah etmeye çalışalım.

Aşağıda (Şekil 35), makinenin her adımda nasıl davranacağı bant üzerinde gösterilmiş ve altında açıklanmıştır. Sarı renge boyalı olan kutular, kafanın o anda üzerinde durduğu bant konumunu temsil etmektedir.



Şekil 35. İkinci örneğin adım adım çalışması

Netice olarak örneğin q1 üzerindeki döngülerden birisi olan b/a,R geçişi, banttan b okunduğunda banta a değerini yaz manasındadır.

2. YAPILAN ÇALIŞMALAR VE BULGULAR

2.1.Giriş

Bu bölümde, M.S.Baptista türü kaotik şifreleme ve DNA kuralları kullanılarak data şifreleme ve deşifreleme gerçekleştirilmiştir. Şifrelemede, lojistik eşitliğin kaotik özelliklerinden yararlanılmıştır. Ayrıca Turing Makinaların özelliklerinden ararlanarak önerilen yöntem modellenmiştir. Lojistik haritanın genel ifadesi şu şekildedir:

$$X_{n+1} = r * X_n(1 - X_n) \tag{8}$$

Uygulamada temel amaç hızlı ve güvenli bir şekilde verileri şifreleyebilen ve güvenli bir şifreleme algoritmasının yerine getirmesi gereken temel nitelikleri ağlayan bir şifreleme algoritması geliştirmektir.

Şifreleme algoritmasının aşağıdaki nitelikleri sağlaması gerekmektedir.

• Gizlilik : Bilgi istenmeyen kişiler tarafından anlaşılamaz.

• Bütünlük : Bilgi saklanması veya iletilmesi sırasında, farkına varılmadan değiştirilemez.

• Reddedilemezlik : Bilgiyi oluşturan ya da gönderen, daha sonra bilgiyi kendisinin oluşturduğunu veya gönderdiğini inkar edemez.

•Kimlik belirleme : Gönderen ve alıcı, birbirlerinin kimliklerini doğrulayabilirler.

Uygulamanın bu şartları sağlaması için kaotik modelle şifrelemeyi DNA şifreleme metodu ile birleştirerek güvenli bir şifreleme modeli oluşturmaya çalışıldı. Uygulamada Gizlilik ve bütünlük şartlarını kaotik modeller şifreleme tekniği kullanarak sağlandı. Yapılan yöntemde guvenliğin daha fazla olması için DNA kurallarından faydalanıldı ve veriler DNA zincirine dönüştürülerek şifrelendirildi .

Kaotik modelle şifreleme de kullanılan başlangıç değerlerini bir anahtar kelimesi olarak kullanıcıdan alınıp ve üzerinde hesaplama yaptıktan sonra şifreleme algoritmasında kullanılındı. Bu; başlangıç değerlerinin açıktan gönderilmesi olayını engellemiş oldu hem de kelime şeklinde olması , kafada tutulması oldukca rahat oldu. Bu

uygulamada işlemler genelde DNA zincirler üzerinde yapılmıştı . Verilerin DNA zincirleri ile bir rasgele DNA zincirleri karşilaştırıldı ve yeni bir DNA zincir üretilir ve böylece şifrelenmiş veri elde edilir. Şifrelenmiş veriler DNA zinciri olarak veya ikili veriye dönüşülmüşü iletile bilir.

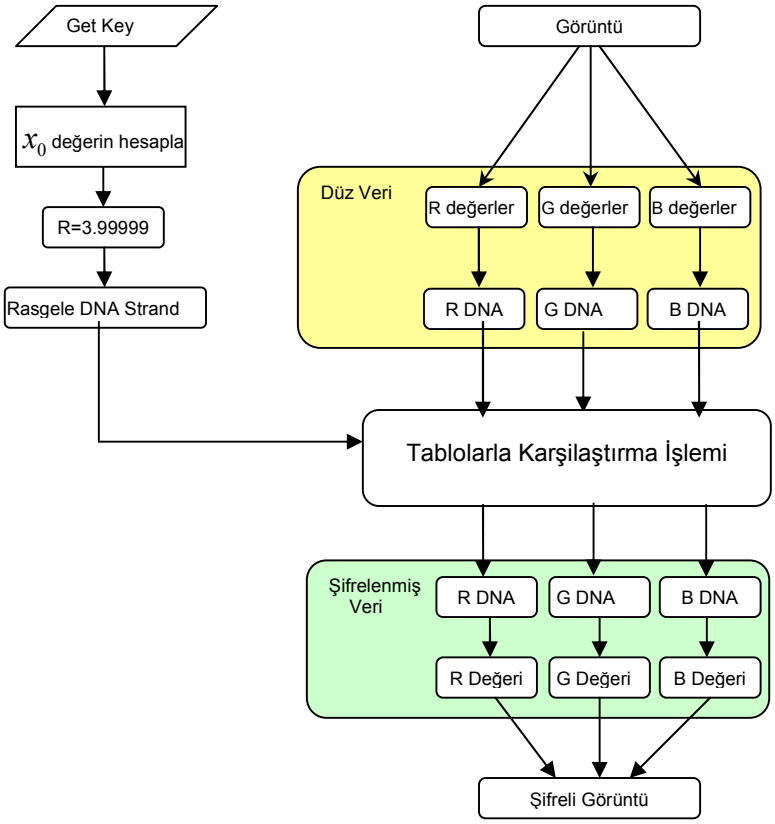
Çalışmada şifrelenmiş DNA zincirler üretilmesi için farklı farklı tablolardan yararlanıldı ve bu tabloları belli zamanlarda değiştirmesi ile guvenliğin daha da artması sağlandı. Yapılan çalışmanın uygun şekilde göstermesi için Turing Makinaları ile modelleme yapıldı.

2.2. Çalışmanın Genel Akış Diyagrami

Genel olarak bu çalışmada bir görüntü DNA zincirlerine dönüştürülür ve elde edilen DNA biçiminde verinin uzunluğunda bir rasgele DNA zincir üretilir. Bu rasgele veriler logistik haritanın kaotik özelliklerinden yararlanarak üretilir ve böylece guvenlik oldukça çoğalır. Daha sonra DNA biçiminde veriler ve rasgele sayılar bir geriye dönüşüm özelliğine sahıp olan tablo yardımı ile yeni bir DNA biçiminde veriye dönüştürülür. Bu yeni veri şifrelenmiş veri olarak kullanılabilir. Şekil 36 da çalışmanın genel akış diyagramı gösterilmiş.

Diyagrama göre yapılan çalışma aşağıdakı bölümlerden oluşmaktadır:

- Görüntünü verilerin DNA biçimine dönüşümü
- Rasgele DNA zinciri üretimi (görüntü boyu *4 uzunluğunda)
- Veri DNA ve rasgele DNA yı karşılaştırıp ve şifreli DNA üretimi
- DNA biçimindeki Şifreli veriyi görüntü şekiline dönüşümü



Şekil 36. Çalışmanın akış diyagrami

Rasgele DNA veriler ile düz metin DNA verilerinin karşılaştırmasında bir geriye dönebilen tablodan faydalanmış. Bu tablodan daha sonra kuralların değiştirerek farklı tablolar üretilir ve belli yerlerde tabloların kullanımı değişilerek , kriptanaliz işleri daha zorlaşır ve yöntemin daha guvenli olması sağlanır.

2.3. Görüntünü Verilerin DNA Biçimine Dönüşümü

Bu çalışmada ilk önce görüntünün piksel değerlerin bir DNA biçimine dönüşümü yapılması gerekiyordi. Dolaysıyla görüntü dosyasından piksel değerlerin birer birer okuyup , DNA verilerine dönüş yapıldı. Bilindiği gibi her piksel 3 farklı değerlerden uluşmuş. Bu değerler sırası ile Kirmizi,Yeşil ve Mavi renklerin miktarların ifade etmektedir. Yapılan çalışmada bu değerleri bir birinden ayrı olarak var sayıp ve bir

görüntüden , 3 farklı DNA zincir oluşturuldu. DNA biçimine dönüş algoritması aşağıda gösterilmiş:

0. StrandR,StrandG,StrandB ← ””
1. for each pixel of Image

    1.1 R,G,B ← Value of Pixel
    1.2 Convert R,G and B to Binary (8 bit)
    1.3 Split each 8 bit to four 2 bit section
    1.4 Convert each 2 bit data to 1 DNA letter {00=’A’,01=’T’,10=’C’,11=’G’}
    1.5 ADD each 4 digite of DNA to StrandR,StrandG and StrandB

Böylece her görüntüden 3 farklı DNA biçiminde veriler üretilir. Bu DNA zincirlerin uzunluğu , görüntünün boyutuna bağlıdır. Örneğin bir görüntünün boyutu m*n ise DNA zincirlerin uzunluğu m*n*4 olmalı. Zira bu görüntü m*n veriye sahip ve her veri 4 DNA harfına dönüşür.

2.3.1. Görüntü DNA Dönüşümü Programı

Görüntüler piksellerden oluşur ve her pikselin 3 farklı değeri var. Yapılan çalışmada ilk önce bu verileri birbirinden ayırıp ve 3 ayrı dizide tutuldu:

```
PlainR=(int *) malloc (PlainSize*sizeof(int));
  PlainG=(int *) malloc (PlainSize*sizeof(int));
  PlainB=(int *) malloc (PlainSize*sizeof(int));
  int i=0;
  for(int j=0;j<Source_image->Width;j++)
    for(int k=0;k<Source_image->Height;k++){
      int PR=(GetRValue(Source_image->Canvas->Pixels[j][k])+i) %256 ;
      PlainR[i]= PR;
      int PG=(GetGValue(Source_image->Canvas->Pixels[j][k])+i) %256 ;
      PlainG[i]= PG;
```

```
            int PB=(GetBValue(Source_image->Canvas->Pixels[j][k])+i) %256 ;
            PlainB[i++]= PB;
          }
```

Tüm piksellerin değerleri her pikselin indeksi ile toplanıp ve 256 modu alınır. Böylece piksellerin değeri , geri dönebilir şekilde değişilir. Bu işlem daha sonra yapılan analizlere göre , şifreleyi daha güvenli hale getirir. Mod 256 işleminin yapılmasının nedeni , her pikselin değerinin 255 den fazla olmamasıdır. Piksellerin görüntüden listeye eklenme işlemide satır satır yapılmıştır. Böyleki önce ilk satırın tüm pikselleri ve sonra ikinci satırın pikselleri görüntüden listeye eklenir ve bu işlem diğer tüm satırlara aynın yapılır.

Görüntünün verileri PlainR,PlainG ve PlainB dizilerinde tutulduktan sonra, bu dizilerin değerleri Decimal biçiminden Binary biçimine ve daha sonra bu biçimden her bayt (8 bit) 4 DNA harfına dönüşmelidi. Dolaysıyla aşağıdakı C de yazılan kodlarla bu işlemler gerçekleşti:

```
PlainStrandR = (char *) malloc (DNAsize*sizeof(char));
  PlainStrandG = (char *) malloc (DNAsize*sizeof(char));
  PlainStrandB = (char *) malloc (DNAsize*sizeof(char));
  CreatePlainDNA(PlainR,PlainSize,PlainStrandR);
  CreatePlainDNA(PlainG,PlainSize,PlainStrandG);
  CreatePlainDNA(PlainB,PlainSize,PlainStrandB);
```

Burada CreatePlainDNA fonksiyonu Plain dizilerin DNA biçimine dönüşümünü sağlar. Bu fonksiyon Kırmızı (PlainR), Yeşil (PlainG) ve Mavı (PlainB) renklere ayrı ayrı yapılır ve sonuçlar sırayla PlainStrandR, PlainStrandG ve PlainStrandB dizilerde tutulur. Aşağıda bu fonksiyonun kodu gösterilmiştir:

```
void CreatePlainDNA(int Plain[],int Length,char PlainStrand[]){
   char temp[]={'A','T','G','C'};
    for(int i=0;i<Length;i++)
       for(int j=0,k=64;j<4;j++,k/=4)
         PlainStrand[i*4+j]=temp[(int) (Plain[i]/k)) & 3];
}
```

Göründüğü koda göre, her sayı 2 bit 2 bit ayrılır ve 2 bitin değeri üzerinden ait oldüğü DNA harfı belli olur. Böylece görüntünun DNA biçiminde verileri PlainStrandR, PlainStrandG ve PlainStrandB dizilerinde tutuldu.

2.4. Rasgele DNA Zinciri Üretimi

Önerilen yöntemde düz metin DNA dizisinin uzunluğunda , başka bir rasgele DNA zincirine ihtiyac var. Bu DNA dizisi , anahtar kelimesine oldukça hessas olmalı ve hakerler tarafından üretilmesi çök zor ve zaman alıcı olmalı. Dolaysıyla yapılan çalışmada , kullanıcıdan bir anahtar kelimesi alarak kaotik haritasinin başlangıç değerin ürettik ve üretilen başlangıç değerle rasgele DNA dizisi oluşturuldu. Kaotik sistemlerin başlangıç değere hessas oldugu , bizim çalışmada avantaj olarak kullanılındı.

2.4.1. Logistik Harıtanın Uygun Katsayısının Bulması

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 x0=0.45 alınırsa,

$$
\begin{aligned}
x1 &= 2 * 0.45 * (1 – 0.45) = 0.49545 \\
x2 &= 2 * 0.49545 * (1 – 0.49545) = 0.499959 \\
x3 &= 2 * 0.\ 499959 * (1- 0.499959) = 0.5 \\
x4 &= 2 * 0.5 * (1 – 0.5) = 0.5
\end{aligned}
\tag{9}
$$

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.

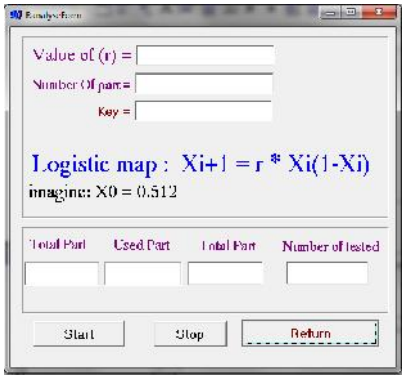
En uygun r bulmak için , aşağıdakı gibi bir çalışma yapıldı. Bu çalışmada 0 ve 1 arasında 256 ayrı ayrı bölgelere kaç iterasyonda tümü isabet etmesi olçüldü.

Kullandığımız yöntem birbirinden farklı 256 rastgele sayı dizisi gerektirdiği ve bu sayıların üretimi lojistik harita ile yapıldığından dolayı, belli bir anahtar ve farklı r değerleri ile bu dizi elemanlarının üretilebilmesi için çeşitli denemeler yapılmıştır. Bu denemelerin sonuçları Tablo 6'da verilmiştir.

Tablo 6. Çeşitli r değerleri ile deneme sonuçları

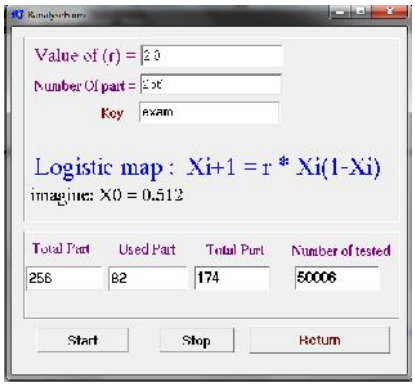
Sayı =256, anahtar = exam			
λ değeri	Bulunan eleman sayısı	Kalan eleman sayı	Deneme sayısı
4.10	59	197	100000
4.05	111	145	100000
4.01	89	167	100000
4.00	256	0	1398
3.99999	256	0	1260
3.9999	256	0	1460
3.999	256	0	1633
3.99	256	0	1858
3.9	256	0	1426
3.8	256	0	2204
3.7	256	0	1564
3.6	256	0	1640
3.5	18	238	100000
2.5	13	243	100000

Tablo 6'daki deneme sonuçlara göre, r değerinin algoritmanın doğru çalışmasını ve hızını etkilediğini söyleyebiliriz. Bu tabloya göre , 3.5 dan aşağı değerlerde 100000 iterasyonun sonuçunda isabet etmiyen bölgeler bile var , ve 256 bölge r= 3.99999 da 1260 iterasyonda tümü isabet edilir ve bu değer bizim uygulamada rasgele DNA zincirler üretmesinde en uygun r değeri sayıldı. Bu tablonu üretmesi için çalışmada şekil 37 da ki gibi bir program hazırlandı.

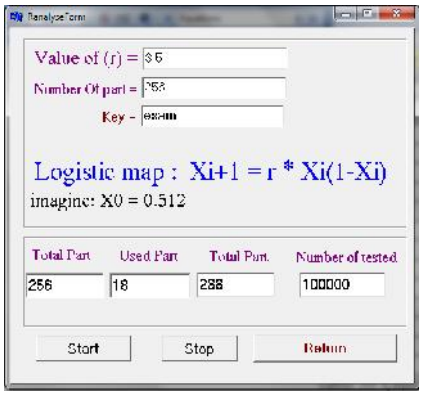


Şekil 37. Farklı bölgelerde üretilen rasgele sayılar için uygun r değeri hesaplama programı.

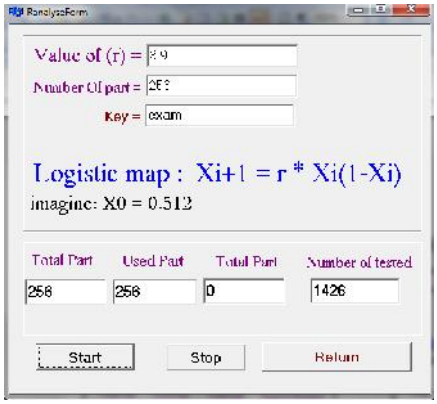
Şekil 37 da gösterilen programda kullanıcı r değeri , bölgelerin sayısı ve başlangıç değeri girerek , programı çalıştırır ve program bu değerlerle logistik haritada uygulur ve iterasyon sayısın (Number of Tested) ve isabet edilen bölgelerin sayısı ve henuz isabet edilmemiş olan bölgeler gösterilir. Örneğin şekil 38 de kaç farklı değerlerle program çalıştırılmış.



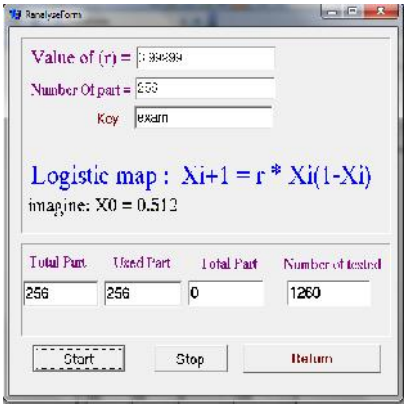
(38.a) r = 2.9



(38.b) r =3.5



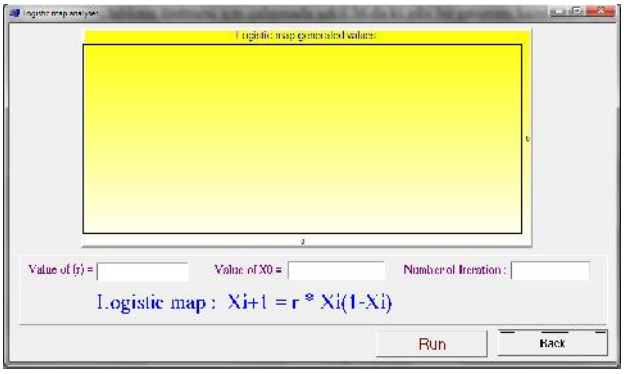
(38.c) r = 3.9



(38.d) r = 3.99999

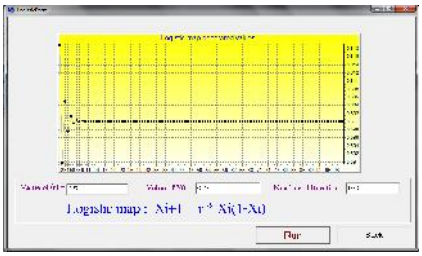
Şekil 38. r değerlerin deneme programının örnekleri

Çalışmada tablo 6 da kı değerler şekil 37 dakı program ile düzenlenmişti. Bu çalışmanın gösterdiğina bir başka kanıt , yaptığımız diğer çalışma da olabilir . Burada yıne bir yeni program hazırlayıp ve görsel olarak çeşitli r değerlerine deneme yapıldı. Şekil 39 de hazırlandığı program gösterilmiş.

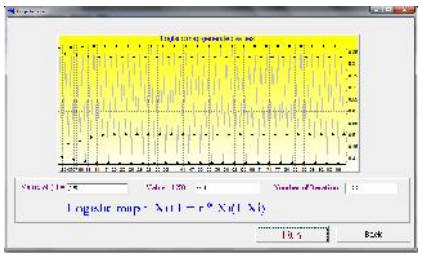


Şekil 39. logistik haritanın inceleme programı

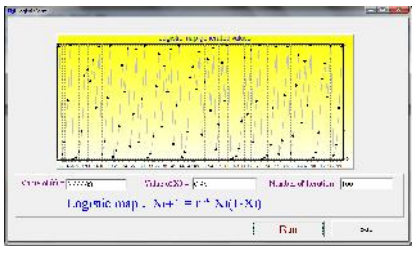
Şekil 39 de gösterilen programda kullanıcı logistik haritanın r , x0 ve ıterasyon sayısın girerek , görsel olarak üretilen rasgele sayılar gösterilir. Şekil 40 ta dört örnek gösterilmiş.



(40.a) . r = 2.5



(40.b) . r = 2.5



(40.c) . r = 2.5

Şekil 40. Logistik haritanın inceleme programının örnekleri

Şekil (40 a) da gösterildiği gibi denklem 19 kanıtlanır . sistem, kararlı duruma sadece 3 iterasyon sonra gelmiştir ve bundan sonraki her iterasyon 0.5 değerini verecektir. (40 b) da sistem düzenli halı var , ancak (40 c) de sistemin ne kadar kaotik olduğu anlaşılır.

Böylece bu çalışmada rasgele sayıların üretiminde , r katsayısının en uygun değeri 3.99999 sayılmıştı.

2.4.2. Rasgele DNA Üretimi

Kaotik özellikleri kullanan şifreleme yöntemleri, genellikle kaotik sistemlerden oluşturulan rastgele sayılar kullanırlar [16,17]. Bu çalışmada önerilen yöntemde lojistik harita aşağıdaki gibi kullanılarak 0 ve 3 arasında rastgele sayılar üretilmiştir:

$X_{n+1} = \lambda X_n (1 - X_n)$, for $X_n \in (0,1)$, and $\lambda \in (3.9996,4]$

Yöntemi görüntü şifreleme işlemlerinde kullanırken λ 'nın değeri 3.99999 olarak seçilmiştir. Elde edilen X_n'ler [0,1] arasında olacağından, bu aralığı 4 parçaya bölmek için bir $\varepsilon = \frac{1}{4}$ parametresi tanımlanmıştır. Böylece [0,1] aralığında bulunan i. parça $((i-1)\varepsilon, i\varepsilon)$ arasında olacaktır. Lojistik haritanın kullanımında başlangıç değeri olarak X_0, algoritmanın anahtarından seçilir. Anahtar kelime, en fazla 80 bitten oluşan bir kelime veya herhangi bir veri olabilmektedir. Bu veriyi 10 ASCII karakteri olarak (her biri 8 bit) $K_0, K_1, K_2, ..., K_9$ biçiminde ifade edebiliriz ve buradaki her bir K_i'yı da 8 bit'ten meydana geldiğinden $K_{i1}, K_{i2}, K_{i3}, ..., K_{i8}$ gibi gösterebiliriz. Başlangıç değeri hesaplama için tüm K lardan aşağıdakı denklem 20 de kullanılındı:

$$X_0 \leftarrow [K_{01} * 2^{79} + K_{02} * 2^{78} + K_{03} * 2^{77} + ... + K_{98} * 2^0] / 2^{80} \quad (10)$$

Anahtar kelimesi 10 karakterden az olabilir. Ancak bu durumda başlangıç değeri karakter sayısına bağlıdı. Anahtar kelimesinin karakter sayısı n ise , başlangıç değer aşağıdakı denklem 21 gibi hesaplanır:

$$X_0 \leftarrow [K_{01} * 2^{n*8-1} + K_{02} * 2^{n*8-2} + K_{03} * 2^{n*8-3} + ... + K_{n8} * 2^0] / 2^{n*8} \quad (11)$$

Çalışmada n*m*4 uzunluğunda rasgele DNA dizisi lazım olması için, bu diziyi lojistik harita yardımı ile üretirilir. Yöntemin şifreleme de deşifrelemede bu diziye bağlı olduğu ve üretilen DNA dizisi başlangıç değere fazlası ile hessas olması , yöntemin temamını anahtar kelimesine bağlı olduğunu belli ediyor. Çalışmada lojistik haritanın parametreleri ($\lambda = 3.99999$ ve X_0) olarak elde edilmiştir.

Bir rastgele DNA listesi oluşturan algoritma aşağıda sunulmuştur:

a) $\varepsilon \leftarrow \frac{1}{4}$
b) $\lambda \leftarrow 3.9999$
c) $i \leftarrow 0$, $k \leftarrow \text{Im}age.Width * \text{Im}age.Height * 4$
d) $n \leftarrow length(key)$
e) $X_0 \leftarrow [K_{01} * 2^{n*8-1} + K_{02} * 2^{n*8-2} + K_{03} * 2^{n*8-3} + ... + K_{n8} * 2^0] / 2^{n*8}$
f) Yeni X_i ($X_{i+1} \leftarrow \lambda X_i (1 - X_i)$)
g) $R \leftarrow$ Yeni X_i in ait olduğu parka ($R\varepsilon, (R+1)\varepsilon$)
h) $Strand \leftarrow Strand + map(R)$ $\quad map(R) = \{A, T, C, G\}$
i) $i \leftarrow i + 1$
j) If i<k goto (f)
k) Return Strand

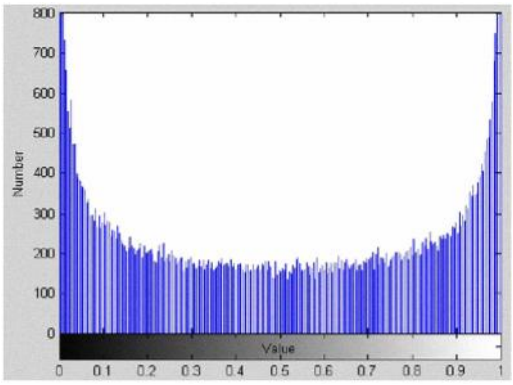
Bu algoritma ile n*m*4 sayılı (0-4 arasında) bir iterasyon listesi oluşturulmuştur. Ancak bu listede {0,1,2,3} değil karşılığı {A,T,C,G} listesinde olan DNA harfları eklenmiştir. Bir sonraki aşamada bu sayılar kullanılarak, bir görüntü verisi şifrelenmiştir. Aşağıdaki C kodu ile bir anahtar kelimeden (KeyStr) bir X_0 değeri hesaplanmaktadır.

```
double Createx0(String KeyStr){
  int n,k=8;  double sum=0;
  n=KeyStr.Length();
  for(int i=1;i<=n;i++,k+=8)
     sum+=(double)KeyStr.operator [](i)*pow(2,k);
  sum+=(double) KeyStr.operator [](1)*pow(2,k);
  k+=8;
  return  sum/pow(2,k); }
```

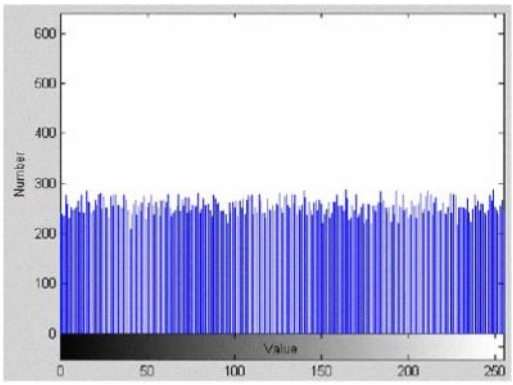
Bu X_0 değeri ile iterasyon listesi oluşturan C kodu ise aşağıda gösterilmiştir.

```
void CreateStrRand(double x0,int Length,char StrRand[]){
    double Landa =3.99999 ;
    int tmp[]={'A','T','G','C'};
    for(int i=0;i<Length;i++){
      StrRand[i]=tmp[(int) (x0*100000000) % 4 ];
      x0=Landa*x0*(1-x0);     }
    return;
}
```

Algoritmaya göre x0 değeri [18] de önerilen gelişmiş kaotik yöntem gibi , kullanmadan önce C katsayısına çarpılır (C=1000000). Şekil 41 de normal kaotik sayıların dağılımı ve şekil 42 de bu dağılımın gelişmiş yöntemle üretilen sayılarda gösterilmiştir.



Şekil 41. Kaotik değerlerin normal dağılımı



Şekil 42. Kaotik değerlerin gelişmiş dağılımı

Böylece CreateStrRand fonksiyonu ile istediğim uzunlukta rasgele DNA zinciri üretebiliriz. Bu dizi daha sonra şifreleme ve deşifreleme işlemlerinde çök önemli rolu olacak.

2.5. Veri DNA ve Rasgele DNA yı Karşılaştırıp ve Şifreli DNA Üretimi

Önceki çalışma bölümlerinde elde edilen rasgele DNA ve Düz veri DNA zincirlirleri bu bölümde bir biri ile karşılaştırılır ve şifrelenmiş DNA zincir elde edilir. Bu işlem bir geriye dönebilen tablo yardımı ile yapılmıştır. Şifrelenmiş DNA zincerleri iki yöntemle elde edilebilir :

- Tek Bir Tablo Yardımı İle
- Çeşitli Tablolar Yardımı ile

2.5.1. Tek Bir Tablo ile Şifreleme İşlemi

Bu yöntemde bir geriye dönebilen tablo yardımı ile Düz metin DNA dizisi ve Rasgele DNA dizisi tablonun satır ve sutunları olarak var sayılırsa , tablonun içerisindeki veriler şifrelenmiş verileri elde ederler. Örneğin tablo 7 de bir geriye dönebilen tablo gösterilmiştir:

Tablo 7. Şifre ve deşifreleme işlemler için bir geriye dönebilen tablo

	T	A	G	C
T	C	G	A	T
A	G	C	T	A
G	A	T	C	G
C	T	A	G	C

Aslında bu tablonun oluşumu çök kolaydır. Eğer T , A , G ve C harfların sıra ile 11 , 10 , 01 ve 00 ikili sayılara karşı var sayalım , bu tablo XOR işlemin temsil eden bir tablodır. Bu tablodan yararlanarak aşağıdakı gibi şifreleme ve deşifreleme işlemleri gerçekleştirilebilir.

2.5.1.1. Tek Bir Tablo ile Şifreleme İşlemi Şifreleme İşlemi

Önerilen yöntemde görüntüden üretilen DNA zinciri ve rasgele DNA zincirlerden birer birer harflar alınır ve görüntü DNA zincirinden alınan satır ve rasgele DNA zincirinden alınan sutun olarak var sayılıp ve bu setır ve sutuna karşı olan harf tablodan seçilip ve şifrelenmiş DNA zincirine eklenir. Bu işlem görüntü DNA zincirindeki tüm harflara tekrarlanır ve elde edilen şifrelenmiş DNA zinciri , görüntünün şifrelenmiş DNA zinciri sayılır. Örneğin görüntü harfı ‘A’ ve rasgele harfı ‘G’ ise , bunlara karşı şifreli harf ‘T’ dir.

2.5.1.2. Tek Bir Tablo ile Şifreleme İşlemi Deşifreleme İşlemi

Önceki bölümde bahs edilen şifreleme yönteminin aynısı deşifreleme işlemine de geçerlidir. Şöyleki görüntü DNA zinciri yerine şifrelenmiş DNA zinciri farz edilirse , şifreleme işlemlerinin aynısının yapılması deşifreleme işleminin gerçekleşmesine neden olabilir. Bunun nedeni tablonun (daha doğru XOR işlemin) geriye dönebilme özelliğine sahıp olmasıdır.

2.5.2. Kriptanaliz Sorunun Araştırması

Bu yöntemde haker veya kriptanaliz , düz metin DNA zincirinin bir kısmını elde ederse, kolaylıkca rasgele diziyi çözebilir ve böylece kriptanaliz işlemleri için önemli veriler elde edebilir. Tabii ki bunun ihtimalı oldukca az, çünkü lojistik harita kaotik özelliklerine sahıpdır. Ancak bu sorunun çözümü ve yapılan şifreleme işlemin daha güvenli olması için ikinci yöntem aşağıdakı gibi önerilindi:

2.5.3. Çeşitli Tablolar ile Şifreleme İşlemi

Bu yöntemde bir tablo değil , çeşitli tablolardan şifreleme ve deşifreleme işlemlerinde yararlanacağız. Bunun için ilk önce bu tür tabloların üretimi ve daha sonra kullanımı tartışılır.

2.5.4. Tabloların Üretimi

Bu bölümün amacı geriye dönebilen özelliğe sahıp olan tabloların üretimidir. Normalde eğer DNA alfabesının tümü için bir 2 bitli ikili sayı var sayılırsa, 2 geriye dönebilen tablo üretilebilir. Bu tabloların biri XOR işlemi ve diğeri – işlemi sonucundan gerçekleşilir. Örnek olarak aşağıdakı homomorfism fonksiyonun var sayalım :

$$\begin{cases} h(T)=11 \quad h(A)=10 \\ h(G)=01 \quad h(C)=00 \end{cases} \tag{12}$$

Bu haritalama sonucunda , tablo 8 ve 9 gibi tablolar üretile bilir.

Tablo 8. XOR işlemi ile ortaya çıkan tablo

XOR	T	A	G	C
T	C	G	A	T
A	G	C	T	A
G	A	T	C	G
C	T	A	G	C

Tablo 9. Eksi işlemi ile ortaya çıkan tablo

-	T	A	G	C
T	C	T	A	G
A	G	C	T	A
G	A	G	C	T
C	T	A	G	C

Bu açık ki , homomorfism fonksiyona farklı miktarlar verilirse , farklı tablolar ortaya çıkabilir. Örneğin :

$$\begin{cases} h(T)=10 \quad h(A)=11 \\ h(G)=01 \quad h(C)=00 \end{cases} \tag{13}$$

Bu haritalama sonucunda , tablo 10 (a ve b) gibi tablolar üretile bilir:

Tablo 10.a . Yeni örneğin XOR işlemi ile ortaya çıkan tablo

XOR	A	T	G	C
A	C	G	T	A
T	G	C	A	T
G	T	A	C	G
C	A	T	G	C

Tablo10.b. Yeni örneğin Eksi işlemi ile ortaya çıkan tablo

-	A	T	G	C
A	C	A	T	G
T	G	C	A	T
G	T	G	C	A
C	A	T	G	C

DNA da 4 alfabe olması nedenınden , $4*3*2*1=24$ farklı durumlar bu homomorfism fonksiyonunda olabilir. Ve her durumu için 2 farklı tablo üretile bilir. Dolayısıyla bu yöntem ile 48 farklı tablolar elde edildi. Böylece yeni yöntemi kullanarak 1 tablo yerine 48 bir birinden farklı tablodan yararlanıp, şifreleme ve deşifreleme işlemleri gerçekleştirildi. Bu tabloların birden fazla olması , şifreleme işlemin dahada karmaşık hale getirir ve kriptanaliz işlemin dahada zorlaştırır.

2.5.5. Tabloların Şifreleme ve Deşifrelemede Kullanımı

Tablo 8 ve 9 a göre her sutunun ait olduğu harf sadece tablonun son satrında yer almış , dolaysıyla her tablonun son satrı , tablolar arasında geçiş yapmağa çök uygun görüştü. Yine tablolara göz attığımızda, satır ve sutunlar aynı olduğunda (örneğin satır 2 ve sutun 2), tabloda yer aldığı harf aslında son satırın hharfıdır. Dolaysıyla şifreleme ve deşifreleme işlemlerinde bu özelliklerden faydalanarak , tablolar arasında geçişler yapabildik ve böylece kriptanalizlere , belli zamanda hangı tablo kullanılmasının anlaşilmasın daha da zorlaştırıldı. Aşağıda bu yöntemin şifreleme ve deşifreleme işlemleri açıklanmıştır:

2.5.6. Çeşitli Tabloların Şifrelemede Kullanımı

Burada bir State gibi değişken elde ederek , ilk önce değerin 1 varsayıldı. Bunun anlamı şu an kullanılan tablo 1. tablodır. Bu durumda şifreleme işlemi , Tek tabloda bahsi giden şifreleme yöntemi gibidir, ancak burada görüntü DNA zincirinden alınan harf (satır için) , tablonun son satırına karşı gelirse (homomorfismde değeri 00 olan harf), state değeri 1 birim artış yapılır ve işlemler devam edilir, ve bu durumda işlemler diğer tablo üzerinden gerçekleşilir. Tabii ki state deüişkenin değeri tabloların sayısından fazlaya gelirse , state deüişkenin değeri 1 olmalı.

Aşağıda önerilen yöntemin C dilinde yazıldığı kodlar açıklanmıştır. Burada tabloların sayısı 48 dir ve ConvertTablesSet() fonksiyonu tüm 48 tabloyu üretir. İlk başta state değişkeni 0 var sayılmış ve (00,01,10,11) e karşı düşen harflerin indeksi 1. tabloda, pos dizisinde tutulur. İşlemler bu pos dizisi üzerinden gerçekleştirilir. Ve uygun zamanlarda state değişkenın , değişmesi ile bu pos diziside güncelleşilir. State değişkeni taboların sayısı (48) den fazla olmaması için , artış zamanlarında aşağıdakı komut kullanıldı :

```
state=(state+1)%48;
```

Böylece state değişkeni 0 ve 47 arasında artır ve 48 olduğu zaman tekrar 0 değeri state değişkeninde tutulur. C dilinde kodlar şu şekilde:

```
void CreateCipherDNA(int PlainStrand[],int StrRand[],int Length,int CipherStrand[]){
        int Pos[4];
        for(int i=0;i<4;i++){
          if(tmp[0][i]==0) Pos[0]=i;
          if(tmp[0][i]==1) Pos[1]=i;
          if(tmp[0][i]==2) Pos[2]=i;
          if(tmp[0][i]==3) Pos[3]=i; }
        ConvertTablesSet();
        for(int p=0,state=0;p<Length;p++){
            CipherStrand[p]= ConvertTables[state][Pos[PlainStrand[p]]][Pos[StrRand[p]]];
            if(Pos[PlainStrand[p]]==3){
                state=(state+1)%48;
                for(int i=0;i<4;i++){
                  if(tmp[state/2][i]==0) Pos[0]=i;
                  if(tmp[state/2][i]==1) Pos[1]=i;
                  if(tmp[state/2][i]==2) Pos[2]=i;
                  if(tmp[state/2][i]==3) Pos[3]=i;
               }
            }
         }
        }
```

CreateCipherDNA fonksiyonun sonunda , şifrelenmiş DNA zinciri üretildi ve CipherStrand dizisinde geri döner.

2.5.7. Çeşitli Tabloların Deşifrelemede Kullanımı

Burada deşifreleme işlemleri , Tek tablodakı deşifreleme işlemlerinin aynısı. Fakat burada satır ve sutun harflar aynın olduğu zaman , state değişkenin değeri artır ve tablolar arasında geçişler gerçekleşir. Şurada deşifreleme C dilinde kodu gelmiştir.

```
for(int p=0,state=0;p<Length;p++){
   PlainStrand[p]=ConvertTables[state][Pos[CipherStrand[p]]][Pos[StrRand[p]]] ;
    if(CipherStrand[p]==StrRand[p]){
        state=(state+1)%48;
        for(int i=0;i<4;i++){
          if(tmp[state/2][i]==0) Pos[0]=i;
          if(tmp[state/2][i]==1) Pos[1]=i;
          if(tmp[state/2][i]==2) Pos[2]=i;
          if(tmp[state/2][i]==3) Pos[3]=i;
        }
    }
}
```

2.6. DNA Biçimindeki Şifreli Veriyi Görüntü Şekiline Dönüşümü

Elde edilen şifrelenmiş DNA zinciri , doğrudan gönderilir veya decimal biçimine dönüşüp bir şifrelenmiş görüntü gibi iletilebilir. Bu bölümde DNA biçiminde verinin görüntü şekiline dönüşü söz konusudur. DNA zincirinden dörder dörder harflar alınıp, eşdeğer ikili değerlere dönüp ve birleşerek 8 bitlik ikili sayı oluşur. Daha sonra bu ikili sayı bir görüntü dosyasına gömülür. İşlemlerin C dilinde kodları aşağıda gelmiştir:

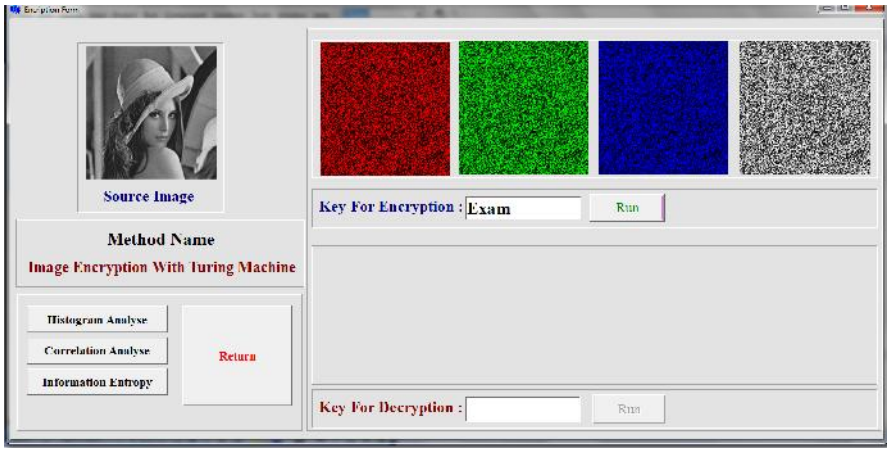
```
void ShowCipher(int R[],int G[],int B[],int size){
for(int col=0,i=0;col< Image1->Width;col++)
  for(int row=0;row< Image1->Height;row++) {
     int Rvalue=R[i+3]+R[i+2]*4+R[i+1]*16+R[i]*64;
     int Gvalue=G[i+3]+G[i+2]*4+G[i+1]*16+G[i]*64;
     int Bvalue=B[i+3]+B[i+2]*4+B[i+1]*16+B[i]*64;
     i+=4;
     Image1->Canvas->Pixels[col][row]=RGB(Rvalue,0,0);
     Image2->Canvas->Pixels[col][row]=RGB(0,Gvalue,0);
```

```
            Image3->Canvas->Pixels[col][row]=RGB(0,0,Bvalue);
            Image4->Canvas->Pixels[col][row]=RGB(Rvalue,Gvalue,Bvalue);
        }
    }
```

Böylece şifrelenmiş 3 DNA zinciri görüntünün R,G ve B değerleri olarak elde edilip ve şifrelenmiş görüntünü gerçekleştirir.

2.6.1. Şifreleme İşlemin Gerçekleştiren Programı

Önerilen yöntem üzerinde bir şifreleme ve deşifreleme programı hazırlandı . bu programda kullanıcı istediği görüntünü seçerek, belli bir anahtar girir. Program girilen anahtar kelimeni kullanarak görüntüyü şifreleyıp ve R,G,B renklerde ve bu renklerin birleşiminden ortayagelen 4 şifreli görüntüyü gösterir . şekil 43 bu işlemin bir örneğin göstermektedir:

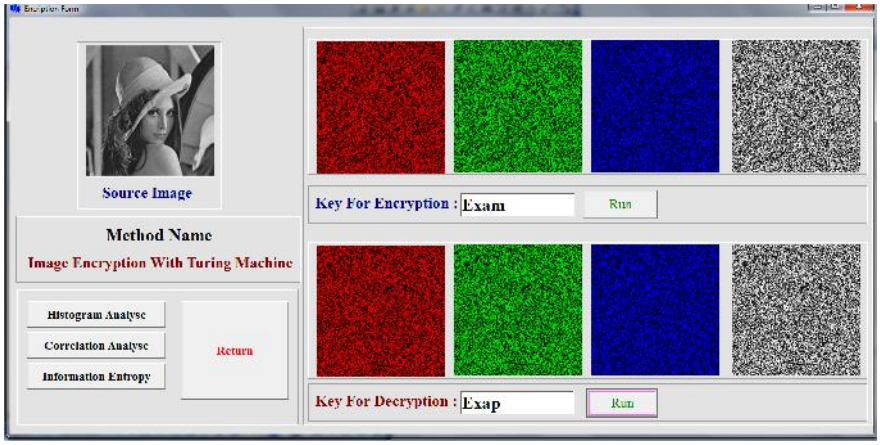


Şekil 43. Şifreleme programindan bir örnek

Deşifre sırasında kullanılan anahtar , şifrelemede kullanılan anahtar ile aynı olduğunda , deşifre işlemi başarılı olacak (şekil 44).

Şekil 44. Deşifreleme işleminin başarılı görüntüsü

Deşifre sırasında kullanılan anahtar , şifrelemede kullanılan anahtar ile aynı değilse , deşifre işlemi başarısız olacak (şekil 45). Bu durumda en ufak değişlik anahtar kelimesinde , sonucu ne kadar etkilediğini dikkata alalım.



Şekil 45. Deşifreleme işleminin başarısızlık görüntüsü

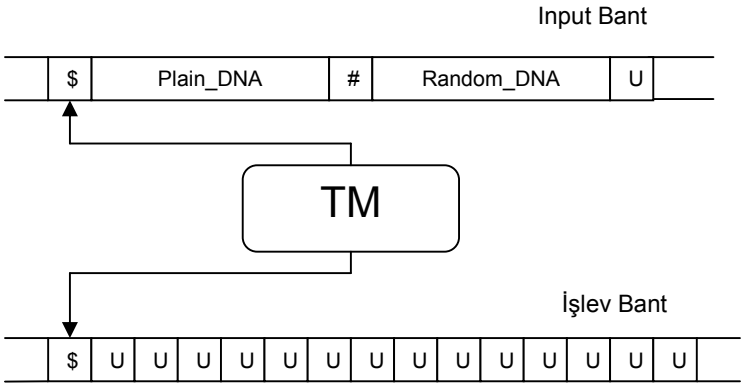
2.7. Yapılan Çalışmanın Modellemesi

Aslında önerilen yöntem ve yapılan işlemleri , Turing Makineler yardımı ile aşağıdakı gibi ifade edebiliriz:

Turing makinenin türü = Tek Teypli (1-tape).

$$\begin{cases} M = (Q, \Sigma, \tau, \delta, q_0, q_{accept}, q_{reject}) \\ \Sigma = \{A, T, G, C, \#\} \\ \tau = \{A, T, G, C, \#, \$\} \end{cases} \quad (14)$$

Problem Tanıtımı: aşağıdakı gibi bir Turing Makinasın varsayalım , bu makina bir input bant ve bir işlev bant'tan oluşmuş. İlk başta input bant'ta Düz görüntünün DNA'ya dönmüş verisi ve daha sonra rasgele DNA zinciri üretilen veriler yer almışlar. (Şekil 46 gibi).



Şekil 46. Problemin başlangıç durumu.

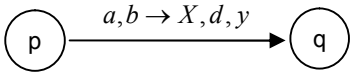
Şekil 46'dakı makineyin İnput Bantı sadece okunabilir ve İşlev Bantı okunma ve yazılma özelliğine sahiptir. Önerilen yöntem için bir model tasarımı söz konusudur.

Bu modellemeni gerçekleştirmek için üç aşamada işlemler .

2.7.1. İlk Aşama : Plain_DNA Verisin İşlev Bantına Aktarması

Bu aşamada, İnput Bant'ta okuma ve İşlev Bant'ta okuma yazma kafalarının '$' simbollara işaret etmesin var sayarak (İnitial Conf.), işlemler yapılacak. Burada İnput Bant'tan birer birer simbollar okunararak , İşlev Bant'ta yazılır. Bu işlem İnput Bant'ta '#' simboluna varmamaya kadar devam edilecek. Tabii ki okunan simbioolar Σ kümesindendir ($\Sigma = \{A,T,G,C\}$).

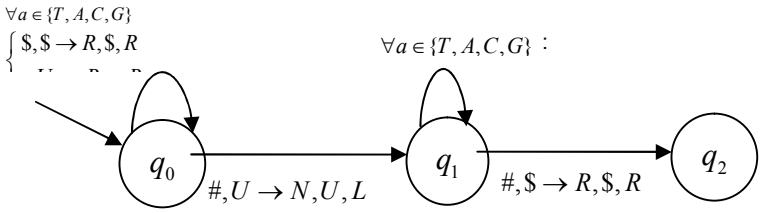
Açıklama: Bu aşamanın modellemesinden önce Transfer Fonksiyonun çeşitli gösterimi ve anlamı açıklanmalıdır. Burada Transfer fonksiyonu ($\delta(p,a,b) = (q,x,d,y)$), şekil 47 gibi gösterilmiş:



Şekil 47. Transfer fonksiyonun modeli

Şekil 47'de makine P durumda ise ve İnput Bant'tan 'a' ve İşlev Bant'tan 'b' simbollerin okuyursa , bir transfer işlemi yapılmalı ve bu transferde İşlevdeki Bant'ta 'b' sembolunun yerine 'd' sembolu yazılıp ve $x, y \in \{L, R, N\}$ Bantların kafa hareketlerin ifade etmektedir (L sağa,R sola ve N hareketsiz durumların belli ediyor).

İlk Aşamada , input Bant'tan Plain_DNA dizisi okunup ve İşlev Banta aktarılmalı, dolaysıyla bu işlemin Turing Modeli şekil 48'de gösterilmiştir:

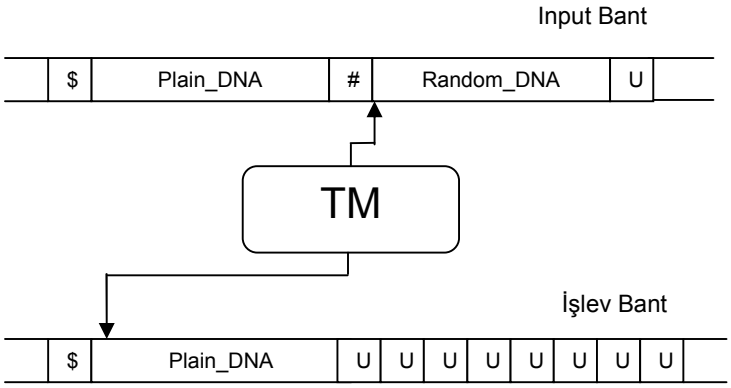


Şekil 48. İlk aşamanın Turing modeli

Şekil 48'deki Turing modelin Transfer fonksiyonu şöyledir:

$\delta(q_0, \$, \$) = (q_0, R, \$, R)$
$\delta(q_0, 'A', U) = (q_0, R, 'A', R)$
$\delta(q_0, 'T', U) = (q_0, R, 'T', R)$
$\delta(q_0, 'C', U) = (q_0, R, 'C', R)$
$\delta(q_0, 'G', U) = (q_0, R, 'G', R)$
$\delta(q_0, \#, U) = (q_1, N, U, L)$
$\delta(q_1, \#, 'A') = (q_1, N, 'A', L)$
$\delta(q_1, \#, 'T') = (q_1, N, 'T', L)$
$\delta(q_1, \#, 'C') = (q_1, N, 'C', L)$
$\delta(q_1, \#, 'G') = (q_1, N, 'G', L)$
$\delta(q_1, \#, \$) = (q_2, R, \$, R)$

İlk aşamanın sonunda Plain_DNA , İnput Banttan İşlev Banta transfer edilir ve Turing Makinenin durumu şekil 49'da gösterilen gibi oluyur:



Şekil 49. İlk aşamanın sonunda makinenin durumu

2.7.2. İkinci Aşama : Şifrelenme İşlemin Gerçekleştirilmesi

Şü aşamada , veriler İnput Bant ve İşlev Bant'tan birer birer okunur ve İşlev Bat'ta Cipher_DNA(şifrelenmiş DNA) yazılır. Bu işlemin yapılmasında bir tablo kullanılır. Önceden bahs ettiğimiz gibi , İnput Bant ve İşlev Bant'tan okunduğu veriler belli tablonun satırı ve sutunu olarak kullanılıp ve tablo'dan alınan veri şifrelenmiş veri olarak İşlev Bant'ta yazılır.

Örneğin tablo 11 bir geriye dönüş tablo var sayalım :

Tablo 11. Geriye dönüş özelliği olan örnek tablo

XOR	T	A	G	C
T	C	G	A	T
A	G	C	T	A
G	A	T	C	G
C	T	A	G	C

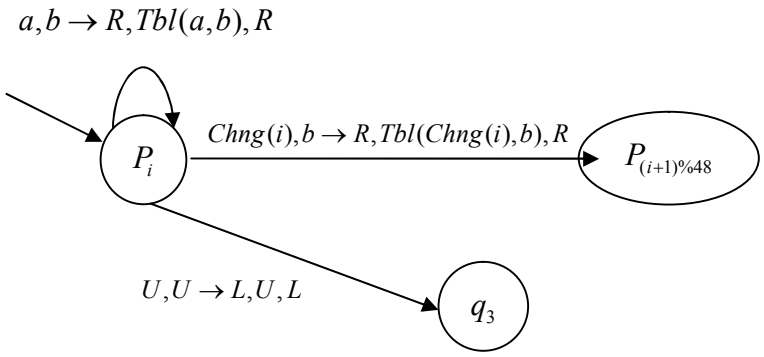
Bu aşamada İnput Bant'tan alan veri tablonun sutunu ve İşlev Bant'tan alınan veri tablonun satırı olarak elde alınır ve tablonun bu satır ve sutununda olan veri , şifrelenmiş veri olarak sayılır.

Tbl gibi fonksiyonu var sayalım. Bu fonksiyonun iki girdi ve bir çıktısı var. Girdiler tablonun satır ve sutunu , cıktı ise girdi tablondakı satır ve sutunun verisidir. Bu işlem denklem (25) ile gösterilebilir:

$$Tbl(a,b) \rightarrow c \tag{15}$$

(Aslında bu fonksiyon bir 2 boyutlu dizi gibi operasyon yapmaktadır.)

Ayrıca i. Sate'in değişimine neden olan simbol Chng(i) var sayılırsa , ikinci aşamanın bir durumunun diyagramı şekil 50'de gösterilmiş:

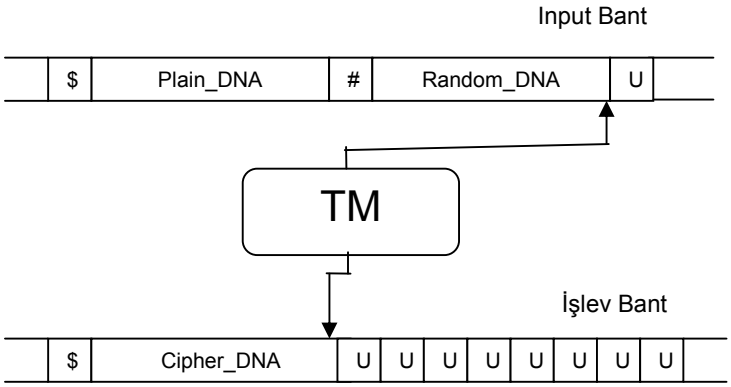


Şekil 50. İkinci aşamanın bir durumunun diyagramı

Bu örnek durum Transfer Fonksiyonu aşağıda gösterilir:

$$\begin{aligned}
&\delta(P_i,a,b) = (P_i,R,Tbl(a,b),R)\\
&\delta(P_i,Chng(i),b) = (P_{(i+1)\%48},R,Tbl(Chng(i),b),R)\\
&\delta(P_i,U,U) = (q_3,L,U,L)
\end{aligned} \tag{16}$$

Böylece tüm veriler Bantlardan okunur ve verilerin şifrelenmiş eşdeğerleri İşlev Bant'ta yazılır. Bu aşamanın sonunda Turing Makinanin genel durumu şekil 51'de gösteriler durumda olmalıdı:



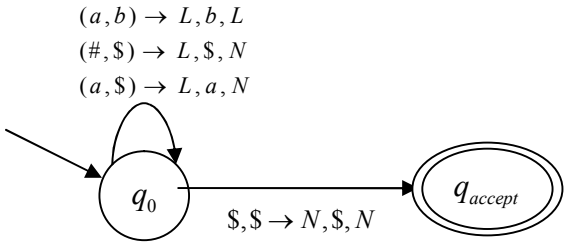
Şekil 51. İkinci aşama sonun makina durumu

Şü durumda şifreleme işlemi tamamlanmış, ve yeter ki İşlev Bantın okuma/yazma kafasın , Bantın başına geri döndürülsün. Bu işlemler üçüncü aşama olarak çalışmada tartışılır.

2.7.3. Üçüncü Aşama : Okuma/yazma kafanın başa dönüşümü

Bu aşamada şifrelenmiş durumda olan makinenin okuma yazma kafasın geriye döüş işlemleri gerçekleştirilir. Burada İnput Bantın kafası sabit olarak , İşlev Bantın kafası birer birer okunup ve verileri değişilmeden sola doğru dönüyür.

Bu aşamanın Turing modellemesi şekil 52'de gösterilmiş:

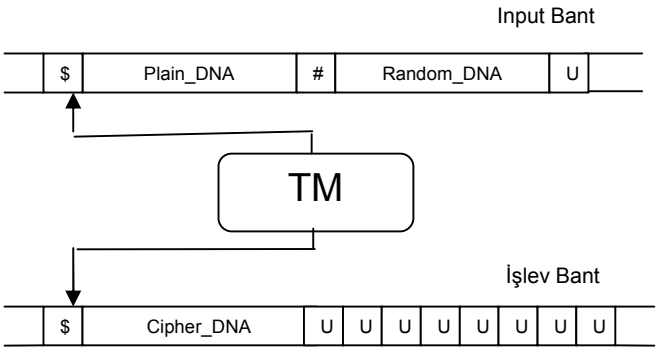


Şekil 52. Üçüncü aşamanın diyagramı

Şekil 52'nin Transfer Fonksiyonu denklem (27) de gösterilmiş:

$$\begin{aligned}
&\delta(q_3, a, b) = (q_3, L, b, L) \\
&\delta(q_3, \#, \$) = (q_3, L, \$, N) \\
&\delta(q_3, a, \$) = (q_3, L, \$, N) \\
&\delta(q_3, \$, \$) = (q_{accept}, N, \$, N)
\end{aligned} \tag{17}$$

q_{accept} durumu , son durum olduğuna göre, makina bu duruma girdiğinde durdurulur ve problem çözülmüş durumuna gelir. Bu durumda makina şekil 53'deki gibi olmalıdır:

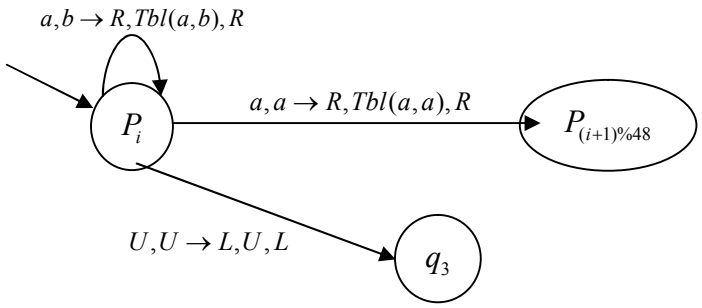


Şekil 53. Makinanın son durumu

Şü aşamanın sonunda, şifrelenmiş DNA veriler İşlev Bant'ta transfer olabilir.

2.7.4. Deşifreleme Modeli

Deşifreleme işlemlerinde Turing Mekinaları ile yapılabilir. Ancak 1. ve 3. aşama şifreleme modelleri ile aynıdır. Sadece 2. aşamada ufak değişikliklar var. Şekil 54'de deşifreleme modelin ikinci aşamasın gösterilmiş:



Şekil 54. Deşifreleme modelin ikinci aşaması

Şekil 54'e göre İnput Bant ve İşlev Bant'tan alınan veriler aynı olduğunda , state'ler arasında geçiş sağlanacak ($a,a \rightarrow R,Tbl(a,a),R$).

Deşifreleme işlemlerin ikinci aşama Transfer Fonksiyonu denklem (28)'de gösterilir.

$$\begin{aligned} &\delta(P_i,a,b) = (P_i,R,Tbl(a,b),R) \\ &\delta(P_i,a,a) = (P_{(i+1)\%48},R,Tbl(a,a),R) \\ &\delta(P_i,U,U) = (q_3,L,U,L) \end{aligned} \tag{18}$$

2.8. Önerilen Yöntemin Güvenlilik Analizleri

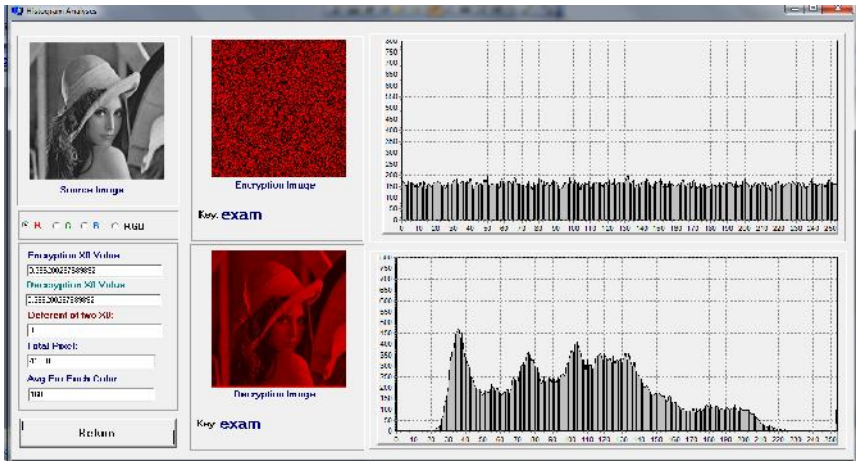
Yöntemin güvenilirliğini göstermek için burada birkaç analiz daha yapılmıştır. Bu analizler üç başlık altında toplanabilir:

- Histogram Analizi
- Korelasyon Katsayısı Analizleri[1]
- Bilgi Entropi[2]

[1] correlation coefficient analyses
[2] Information Entropy

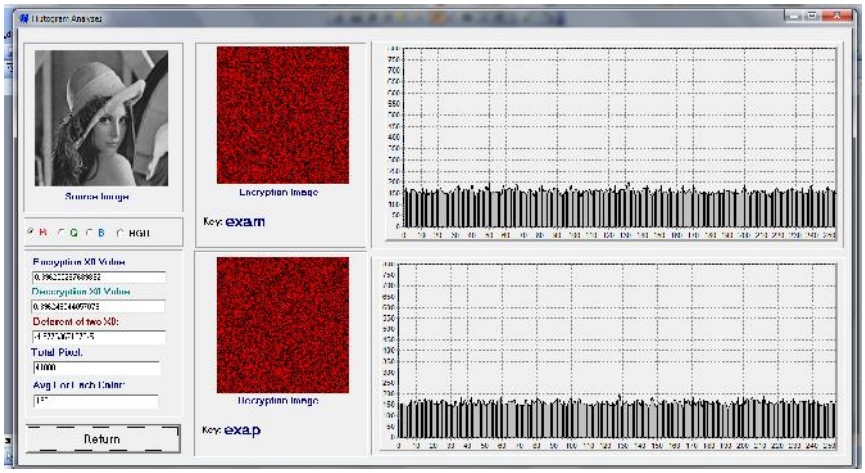
2.8.1. Histogram Analizi

Düz görüntü ve şifrelenmiş görüntünün histogramı Şekil 55'da gösterilmektedir (görüntülerin kırmızı renkleri seçilmiştir). Burada gösterilen düz görüntü renkli olduğundan dolayı, histogramlar hem düz hem de şifreli görüntüler üzerindeki kırmızı, yeşil ve mavi renk dağılımına ayrıca bunların berleşim renklerine göre yapılmıştır . Şekil 55'ye göre düz görüntülerin histogramının istatistiksel analize ne kadar elverişli olduğu ve önerilen yaklaşımın istatistiksel analize karşı ne kadar sağlam durduğu açıkça görülmektedir.



Şekil 55. Başarılı şifreleme'de Histogram analizi

Bir başka çalışmada şifreleme ve deşifreleme anahtar kelimelerin , bir birinden farklı girildi. Anahtar farkı bu iki kısmda oldukca çök ufak tutulmuş. Şekil 56'de göründüğü gibi , bu kuçuk değişiklik anahtar kelimede ne kadar Histogram Analizinde uygun olduğu belli oluyur.

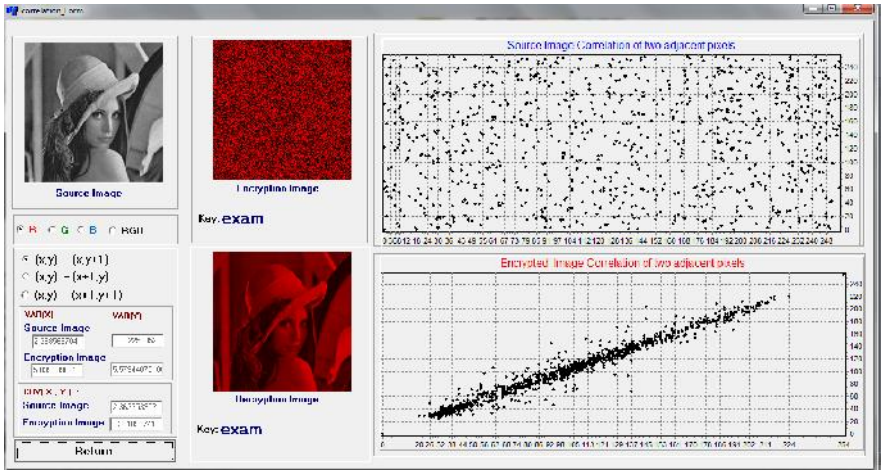


Şekil 56. Başarısız şifreleme'de Histogram analizi

Şekil 56'de şifreleme anahtarı 'exam' ve deşifreleme anahtarı 'exap' var sayılmıştır. Ancak göründüğü gibi, Histogram Analizi her iki görüntüde iyi renk dağılımlarını belli etmektedir. Böylece kriptanalizler ne kadar anahtar kelimesine yakınlaşıp , uzak oldukların anlaşamazlar.

2.8.2. Korelasyon Katsayısı Analizleri

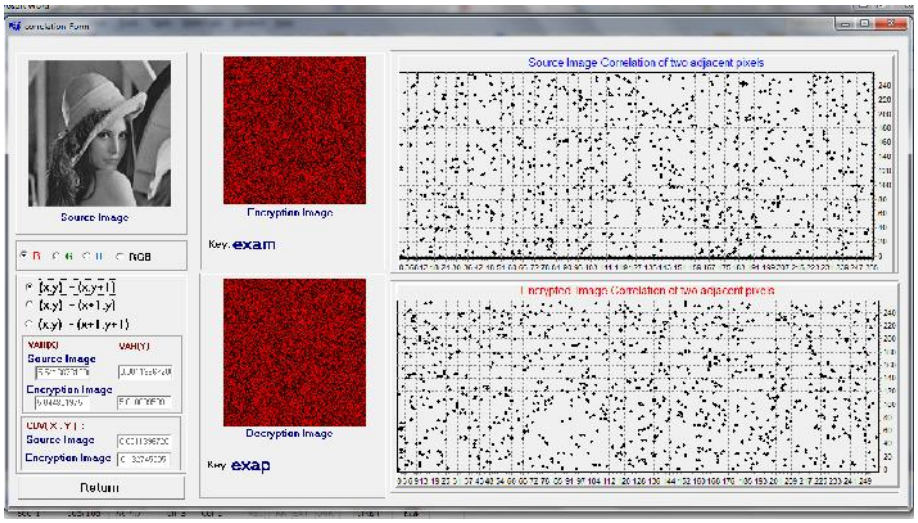
Basit korelasyon analizi, iki değisken arasındaki ilişkinin düzeyini (derecesini-siddetini-gücünü) ve yönünü belirlemek amacıyla yapılır. Her iki değişkenin de sürekli değişken olması ve değişkenlere ilişkin verilerin normal dağılım göstermesi durumunda değişkenler arasındaki ilişki Pearson korelasyon katsayısı ile belirlenir. Korelasyon katsayısı ile belirlenen ya da ölçülen, söz konusu değişkenler arasındaki doğrusal ilişkidir. Eğer değişkenler arasındaki ilişki doğrusal değil ise hesaplanan korelasyon katsayısı değişkenler arasındaki ilişkiyi ölçmek için uygun bir istatistik değildir. Burada görüntülerin komşu pixel'ler arasında doğrusal ilişkilerinin olup olmadığını belirlemek için Şekil 57'de gösterilen analiz yapılmıştır .



Şekil 57. Başarılı şifreleme'de Korelasyon Katsayısı analizi

Şekil 57'e göre düz görüntüde bu ilişki doğrusal ve şifreli görüntüde ise doğrusal değildir. Bundan dolayı, yapılan şifreleme işleminin istatistiksel analizlere kapalı olduğu sonucunu kolayca çıkarabiliriz.

Diğer bir deneyimde , önceki analiz gibi , şifreleme anahtar ile deşifreleme anahtarı birbirinden farklı alarak, programı köştürdük. Şekil 58'da gösterilen gibi deşifre görüntü orjinal görüntüden daha farklı oldu. Göründüğü gibi ilişkiler ise deşifre olmuş görüntüde , şifreli görüntüye benzer , doğrusal değil.



Şekil 58. Başarısız şifreleme'de Korelasyon Katsayısı analizi

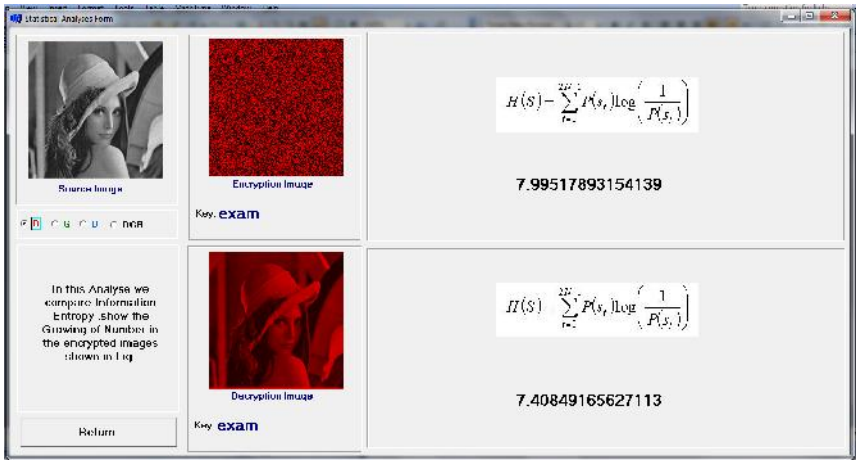
Şekil 58'da şifrelemede kullanılan anahtar kelime 'exam' ve deşifrede kullanılan anahtar kelime ise 'exap' var sayılmış. Dolayısıyla bir ufak değişim anahtar kelimelerde deşifrelemenin ne kadar başarısız olduğu soz konusu. Böylece kriptanalizler ne kadar anahtar kelimesine yakınlaşıp , uzak oldukların anlaşamazlar.

2.8.3. Bilgi Entropisi

Bilgi teorisinde, entropy rastgele sayılar arasında belirsiz bir ilişkiyi bulmak demektir. Bu terim aslında **Shannon entropy'**sine dayandırılmıştır ve kısaca aşağıdaki denklem ile ifade edilebilir :

$$H(S) = \sum_{i=0}^{2N-1} P(s_i) \log\left(\frac{1}{P(s_i)}\right) \tag{19}$$

Bu ifadenin küçük değerler üretmesi istatistik analizlerde kullanımının daha uygun olacağı anlamına gelir. Bir şifrelenmiş görüntünün başka yönlerden güvenli olup olmadığını araştırmak için, bilgi entropy'sinden yararlanabiliriz. Bunun için her görüntünün pixel değerleri P ile temsil edilip görüntü boyutu da N*N alınırsa, denklem (29) düz ve şifreli görüntülerin her biri için H(S) değerlerini hesaplayabilir. Şekil 59'de bu değerler düz ve şifrelenmiş görüntüler için gösterilmiştir.



Şekil 59. Başarılı şifreleme'de Bilgi Entropi analizi

Denklem 19'un verdiği sonuç ne kadar 8 sayısına yakın ise , o kadar kriptanaliz tahdidlerine karşı dayanaklıdır. Şekil 59'deki verilere göre şifrelenmiş görüntüden elde edilen sonuc oldukca 8'e yakın ve düz görüntünün sonucu daha kucuk ve 8'den daha uzak , dolaysıyla önerilen yöntemin başarılı olduğunu ortaya koymaktadır.

Önceki analizler gibi , bu analizde de bir başka deneyımde , anahtar kelimeleri farklı kullanıldı. Elde edilen sonuclar , önerilen yöntemin ne kadar anahtar kelimesine bağlı ve hassas olduğunu kanıtlır. Şekil 60'de bir başarısız şifreleme işlemin Bilgi Entropi analizin göstermektedir.

Şekil 60. Başarısız şifreleme'de Bilgi Entropi analizi

Şekil 60'de verilen görüntüler düz bir görüntünü 'exam' anahtarı ile şifreleyıp ve 'exap' anahtarı ile deşifrelemesinden elde edilmiştir. Böylece kriptanalizler ne kadar anahtar kelimesine yakınlaşıp , uzak oldukların anlaşamazlar.

2.8.4. Diğer Çalışmalarla Karşılaştırma

Bu bölümde çalışmanın ne kadar iyi olması tartışılır. Kriptografı çalışmalarda genelde Histogram ve Güvenilik analızleri başarılı olması yeterlidir. Bunun nedeni, her şifreleme yöntemi zaman içerisinde güvenli durumdan güvensiz duruma geçmesidir. Her yötemin önerildiğinden bir süre sonra yöntemin çözülebileceği köşkülü olabiler. Dolaysıyla yeni yöntemler önerilir ve başarılı veya başarısız olması olçulur. Ancak tablo

12 de gösterildiği gibi, yapılan çalışmanın şifrelenmiş görüntünün Bilgi Entropiden elde edilen sonuç üç önlü yöntemler ile karşılaştırılmıştır.

Tablo 12. Yapılan çalışma ve diğer çalışmaların Bilgi Entropi karşılaştırılması.

Ideal	8
Baptista's	7.926
Wong's	7.969
Xiang's	7.995
RC5	7.9884
RC6	7.9899
Wei-bin et al.	7.9971
Önerilen yöntem	7.9959

Tablo 12 deki gibi yapılan çalışmadan elde edilen sonuç diğer yöntemlere göre daha iyidir.

3. SONUÇ

Bu çalışmada bir görüntü şifreleme yöntemi önerilmiş olup, yöntemin güvenliği düz görüntüler ile şifrelenmiş görüntüler arasında gerçekleştirilen dönüşümler göz önünde bulundurularak analiz edilmiştir. dolayısıyla kullanılan anahtar kelimenin daha hassas olduğu söylenebilir. Önerilen yöntemde DNA özellikleri göz önüne alınarak, ve bir rasgele DNA zinciri yararlanarak, şifreleme işlemleri gerçekleşmiştir. Şifreleme ve deşifreleme yönteminde bir geriye dönebilen tablo ile işlemler gerçekleşmiştir. Ancak kriptanalize daha dayalı olmak için bir tablo yerine 48 tablo kullanılındı. Bu tablolar belli zamanlarda, düz ve rasgele verilere bağlı, bir birine geçiş yapabilirler. Bu tablolar arasında geçişler kriptanalizi daha zorlaştırabilir. Ayrıca şifrelenmiş veriler üzerinde yapılan analizlere göre bu yöntem güvenliliğida sağlanılmıştır.

Bu çalışmada, algoritma açık olduğuna göre , kriptanalizi dahada zorlaştırması için iki açıdan yaklaşmıştı. Bunlardan birisi kaotik sistemlerin özelliklerinden yararlanarak rasgele diziler üretip ve bu dizileri şifrelemede kullanmasıdı. Böylece algoritma oldukca anahtar kelimeye bağlı ve hassas olmaktadır. Diğer yönden algoritmada 48 farklı tablo kullanılması , yinede ne zaman hangı tablo işlem yapması tamaman düz ve rasgele verilere bağlı olmuştı. Dolaysıyla kriptanaliz , rasgele dızı veya düz metini bile elde ederse, yine hangı tabloyı kullanmasın anlamayacaktır.

Bu tür yaklaşımlar şifreleme yöntemlerinde , farklı algoritmalardada yapılabilir.

4. KAYNAKLAR

[1] http://www.byte.com.tr Dosya Şifreleme Programları. 01 Subat 2006.

[2] http://www.btguvenlik.telekom.gov.tr Bilgi Güvenliği. 13 Mayıs 2006.

[3] Bozkurt, F., Elektronik Güvenlik, Şifreleme Teknikleri ve Algoritması Açık Olan Şifreleme Teknikleri ,Public Key Encryption, Kasım 2005, İzmir.

[4] Konheim, A., Cryptography: A Primer, 1981 , New York Wiley.

[5] Feistel H., Cryptography and Computer Privacy, Scientific American, 228,5 (1973) 15-23.

[6] Eren, A., M., Açık Anahtarlı Kriptografi, Penguence, 3 (2005) 28-32.

[7] http://www.simetri.com Sayısal İmza Nedir?. 01 Mayıs 2006.

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

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

[10] Li, S., When Chaos Meet Computers, Elsevier Science, 4 (June 2004) 145-149.

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

[12] Alvarez, G. and Li, S., Some Basic Cryptographic Requirements for Chaos-Based Cryptosystems, International Journal of Bifurcation and Chaos, 16,7 (May 2005) 125-128.

[13] Kangwei, H. and Lih, T., C.,Parwani, Chaos and Cryptography: Applications and Analysis, Mart 2003 , Singapore.

[14] Solak, E., Cryptanalysis of Observer Based Discrete-Time Chaotic Encryption Schemes, International Journal of Bifurcation andChaos, 15(2005) 62-66.

[15] Solak, E., On the Security of Discrete-Time Chaotic Cryptosystems, Physics Letters A, 320 (2004) 389-395.

[16] Pisarchik AN, Flores-Carmona NJ, Carpio-Valadez M. Encryption and decryption of images with chaotic map lattices. Chaos, Interdiscipl J Nonlinear Sci2006,16 (2003) 33-38.

[17] Fridrich J. Symmetric ciphers based on two-dimensional chaotic maps. Int J Bifurcat Chaos, 8 (1998) 1259–1284.

[18] Qian W., Qiang Z., Xiaopeng W. Image Encryption Algorithm based on DNA Biological Properties and Chaotic Systems, Ekim 2004, Dalian China.

ÖZGEÇMIŞ

1981 yılında İRAN'ın Tabriz şeherinde dogdu. İlkögretimini Parvaz İlkokulu, orta ögretimini Almahdi Ortaokulu ve Seddige Kobra Lisesi'nin matematik Agırlıklı bölümünde tamamladı. 2000 yılında Tabriz AZAD Üniversitesi, ana bilimler Fakültesi, Uygulamalı matematik Bölümü'nde okumaya hak kazandı. Bu bölümden, 2005 yılında mezun oldu. 2005 yılından itibaren, Tabriz'in Teknik Liselerinde bilgisayar öğretmeni olarak görev yaptı. 2009 yılında, Karadeniz Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Ana Bilim Dalı'nda Yüksek Lisans Programına başladı.