KAOS TABANLI BİR GÖRÜNTÜ ŞİFRELEME ALGORİTMASI VE DONANIMSAL ORTAMDA GERÇEKLEŞTİRİLMESİ

Yük. Müh. Hidayet OĞRAŞ

Doktora Tezi
Elektrik-Elektronik Mühendisliği Anabilim Dalı
Doç. Dr. Mustafa TÜRK

TEMMUZ-2017

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

KAOS TABANLI BİR GÖRÜNTÜ ŞİFRELEME ALGORİTMASI VE DONANIMSAL ORTAMDA GERÇEKLEŞTİRİLMESİ

DOKTORA TEZİ

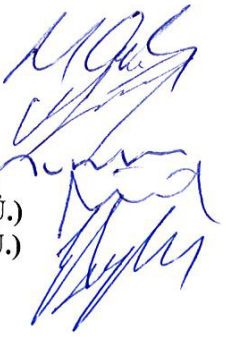
Yük. Müh. Hidayet OĞRAŞ

(101113202)

Tezin Enstitüye Verildiği Tarih: 19.06.2017
Tezin Savunulduğu Tarih: 21.07.2017

Tez Danışmanı: **Doç. Dr. Mustafa TÜRK (F. Ü.)**

Diğer Jüri Üyeleri: **Doç. Dr. Arif GÜLTEN (F.Ü.)**
Doç. Dr. Taner TUNCER (F.Ü.)
Yrd.Doç.Dr. Mehmet ÜSTÜNDAĞ (Bingöl Ü.)
Yrd.Doç.Dr. Erdinç AVAROĞLU (Mersin Ü.)



TEMMUZ-2017

ÖNSÖZ

Bu tez çalışmasında sayısal görüntülerin şifrelenmesine yönelik olarak hızlı ve güvenli bir şifreleme-çözme yapısı sunulmuştur. Şifreleme anahtarlarının üretiminde yeni bir kaotik haritadan faydalanılmış ve diferansiyel saldırılara karşı yüksek seviye güvenlik sağlaması için kaynak görüntüye hassas duyarlılık sergileyen bir şifreleme algoritması tasarlanmıştır. Sunulan kripto sisteme yönelik olarak bu alanda sıkça kullanılan tüm istatistiksel ve diferansiyel analizler incelenmiş olup, bilinen bazı şifreleme algoritmaları ile sistemin performans karşılaştırması yapılmıştır. Şifrelenmiş görüntü iletiminde olabilecek veri kaybı ve gürültü saldırısı altındaki performansının tatmin edici seviyede olduğu kripto sisteme ait teorik sonuçlarla değerlendirilmiştir. Ayrıca uygulama noktasında, sunulan şifreleme algoritmasının sahada programlanabilir kapı dizilerinde gerçekleştirilmesi tüm sistemin donanım ortamında sorunsuz çalışabileceğini göstermiştir.

Bu tez çalışmasının başlatılmasında, geliştirilmesinde ve tamamlanmasında gerekli sabrı, özveriyi, tavsiyeleri ve desteğini esirgemeyen değerli danışmanım Sayın Doç. Dr. Mustafa TÜRK' e teşekkürlerimi sunarım.

Tez çalışmalarımda değerli fikirleri ve tecrübesiyle bana her aşamada yol gösteren çok sevdiğim babam, Prof. Dr. Sezai OĞRAŞ' a teşekkür ederim.

Doktora eğitimim sırasında gösterdiği sürekli sabır, sevgi ve özverisiyle her zaman yanımda olan ve beni her konuda destekleyen eşime teşekkür ederim.

HİDAYET OĞRAŞ

Elazığ-2017

İÇİNDEKİLER

ÖZET

Bu tezde, renkli ve gri görüntüler için kaos tabanlı simetrik bir görüntü şifreleme algoritması tasarlanmıştır. Sunulan algoritma, görüntü pikselleri ile ilgili permütasyon (karıştırma) ve değiştirme (dağılma) işlemlerini içermektedir. Karıştırma kısmında kaynak görüntünün piksel pozisyonlarını yer değiştirmek için Arnold kedi haritası kullanılmış ve ardından dağılma kısmında ise şifreleme kodlarının üretildiği iyileştirilmiş Lojistik harita sistemi piksel değerlerini değiştirmede kullanılmıştır. Öncelikle parametre alanı, karmaşıklık, tekdüzelik ve kaotik zenginlik bakımından standart Lojistik haritaya göre çok daha iyi performans sergileyen iyileştirilmiş bir kaotik sistem tasarlanmıştır. Bu sistemden üretilen şifreleme kodları sadece sistemin kontrol parametresine ve başlangıç durumuna bağlı olmadığı gibi aynı zamanda kaynak görüntü karakteristiğine de hassas duyarlıdır. Dolayısıyla sunulan yapı istatistiksel saldırılar, diferansiyel saldırılar, bilinen-bilgi ve seçilen-bilgi saldırı tekniklerine karşı dayanıklıdır. Ayrıca sistemden yüksek şifreleme gücü elde etmek için karıştırma ve dağılma işlemleri birden fazla tekrarlanmakta ve her iterasyonda farklı şifreleme kodları dağılma kısmında kullanılmaktadır.

Sunulan görüntü kripto sistem ile ilgili tüm istatistiksel ve diferansiyel analizler detaylı bir şekilde incelenmiştir. Hız ve güvenlik açısından sunulan algoritma, mevcut bazı şifreleme algoritmaları ile karşılaştırıldığında tasarlanan algoritma çok daha iyi performans göstermiştir. Sunulan kripto sistemin uygulama tarafında ise anahtar üreteci, şifreleme ve çözme yapıları donanım olarak sahada programlanabilir kapı dizileri (FPGA) ortamında başarıyla sentezlenmiş ve gerçekleştirilmiştir. Teorik analizler ve benzetim sonuçları, sunulan görüntü şifreleme sisteminin yüksek güvenlik seviyesine sahip olduğunu doğrulamakta ve farklı ölçülerdeki karasel görüntüler üzerinde etkin bir şifreleme ve çözme işlemi gerçekleştirdiğini onaylamaktadır. Ayrıca deneysel sonuçlar, sunulan kripto sistemin pratik uygulamalarda kullanılabileceğini göstermektedir.

Anahtar kelimeler: Kaos, Lojistik harita, Kriptografi, Görüntü işleme, FPGA, Şifreleme, Arnold Kedi Haritası

SUMMARY

A Chaos-based Image Encryption Algorithm with its Hardware Implementation

In this thesis, a chaos-based symmetric image encryption algorithm is designed for color and gray images. The proposed algorithm contains permutation (confusion) and modification (diffusion) processes related to the image pixels. In confusion part, Arnold Cat map is used to change the pixel positions of the plain image and then improved Logistic map for the generation of encryption key is used to modify the pixel values in diffusion part. Firstly, an improved chaotic map is designed to exhibit better performance than the standard Logistic map in terms of key space range, complexity, uniformity and chaotic substantiality. Generated encryption codes are not only sensitive to the control parameter and initial condition of the improved map but also strongly depend on the plain image characteristic. Hence, the proposed scheme can resist statistical attacks, differential attacks, known-plaintext and chosen-plaintext attacks. Furthermore, to get higher encryption strength of the cryptosystem, both confusion and diffusion processes are repeated one more time and different encryption codes are used in each iteration in diffusion part.

All statistical and differential analyses about the proposed cryptosystem are studied in detail. Compared to the results of the some conventional encryption algorithms, the proposed scheme shows superior performance according to the speed and security of the algorithm. The practical applications including key generator, encryption and decryption parts of the proposed cryptosystem are successfully synthesized and implemented into the FPGA (Field Programmable Gate Array) as a hardware environment. Theoretical analyses and simulation results confirm that the proposed image encryption system has high level of security and effectively encrypts and decrypts the square images with different sizes as well. Experimental results also demonstrate the feasibility of the proposed cryptosystem in practical applications.

Keywords: Chaos, Logistic Map, Cryptography, Image processing, FPGA, Encryption, Arnold Cat Map

ŞEKİLLER LİSTESİ

Sayfa No

TABLOLAR LİSTESİ

Sayfa No

SEMBOLLER LİSTESİ

$\boldsymbol{\alpha}$	**:** İLH sistem parametresi
$\boldsymbol{\beta}$	**:** Gecikme
$\mathbf{b}_{\boldsymbol{n}}$	**:** Bit akışı serisi
$\mathbf{p}(\mathbf{x}_{\boldsymbol{i}})$	**:** x_i sembolünün olasılığı
$\mathbf{r}_{\boldsymbol{d}}$	**:** d gecikmesindeki çapraz korelasyon katsayısı
$\mathbf{x}_{\boldsymbol{n}}$	**:** Durum değişkeni
$\mathbf{A}$	**:** Kripto sistemin anahtar alanı
$\mathbf{B}$	**:** Olası toplam kod sayısı
$\mathbf{C}$	**:** Şifrelenmiş görüntü
$\mathbf{D}$	**:** Çözme fonksiyonu
$\mathbf{E}$	**:** Şifreleme fonksiyonu
$\mathbf{H}(\mathbf{x})$	**:** x üretecinin entropisi
$\mathbf{K}$	**:** Şifreleme kodu
$\mathbf{M}, \mathbf{N}$	**:** Görüntü ölçüsü
$\mathbf{N}$	**:** Karasel görüntü ölçüsü
$\mathbf{P}$	**:** Kaynak görüntü
$\mathbf{R}$	**:** Orijinal görüntü
$\mathbf{T}$	**:** Eşit piksellerin sayısı
$\mathbf{U}$	**:** Karışmış görüntü
$\mathbf{cc}$	**:** Benzerlik katsayısı
$\mathbf{e}$	**:** Hata sinyali
$\mathbf{i}, \mathbf{j}$	**:** Orijinal görüntüdeki piksel koordinatı
$\mathbf{n}$	**:** İterasyon
$\mathbf{p}, \mathbf{q}$	**:** AKH parametreleri
$\mathbf{pk}$	**:** Kaynak görüntü karakteristiğiyle ilişkili parametre
$\boldsymbol{\lambda}$	**:** Lyapunov değeri
$\boldsymbol{l}$	**:** Piksel değerinin temsil edildiği bit sayısı
$\mathbf{K_1}, \mathbf{K_2}$	**:** Şifreleme kodları
$\boldsymbol{\chi}^2$	**:** Chi-karasel değeri
$\mathbf{r}$	**:** SLH sisteminin kontrol parametresi
$\mathbf{s}$	**:** İLH sisteminin kontrol parametresi
$\mathbf{w}, \mathbf{v}$	**:** Karışmış görüntüdeki piksel koordinatı
$\mathbf{x}, \mathbf{y}$	**:** Piksel koordinatı

KISALTMALAR LİSTESİ

AES : Gelişmiş Şifreleme Standardı
DES : Veri Şifreleme Standardı
IDEA : Uluslararası Veri Şifreleme Algoritması
OTP : Tek Kullanımlı Anahtar
PRNG : Sözde-Rastgele Sayı Üreteci
AKH : Arnold Kedi Haritası
SLH : Standart Lojistik Harita
İLH : İyileştirilmiş Lojistik Harita
FIPS : Federal Information Processing Standards
NIST : National Institute of Standards and Technology
RGB : Kırmızı, Yeşil, Mavi
OHM : Ortalama Hareket Mesafesi
EPSO : Eşit Piksel Sayılarının Oranı
MAE : Ortalama Mutlak Hata
NPCR : Number of Pixels Change Rate
UACI : Unified Average Changing Intensity
PSNR : Peak Signal to Noise Ratio
MSE : Ortalama Karasel Hata
EQ : Şifreleme Kalitesi
HD : Hamming Mesafesi
FPGA : Sahada Programlanabilir Kapı Dizileri
CLB : Configurable Logic Block
VHDL : Very high speed integrated circuit Hardware Description Language
VGA : Video Grafiği Dizisi
CRT : Cathode Ray Tube
LED : Light Emitting Diode
ISE : Integrated Synthesis Environment
XSG : Xilinx System Generator

1. GİRİŞ

1.1. Genel Bilgiler

Bilgi teknolojisindeki gelişimin ve bilgisayar ağlarının hızlı büyümesi ile birlikte güvenli olmayan kanallar üzerinden iletilen sayısal veri miktarı her geçen gün artmaktadır. Bununla birlikte haberleşme sistemlerine duyulan ihtiyacın artması ve multimedya uygulamalarının geçmişe göre çok daha yaygın kullanılması sonucu bilginin güvenli ve doğru bir şekilde iletilmesi ciddi bir sorun olmaya başlamıştır. Özellikle internet üzerinden yaygın bir şekilde kullanılan elektronik yayıncılık ve multimedya verilerinin yasadışı çoğaltılması ve dağıtımı önemli bir problem haline gelmiştir [1]. Bu durum, haberleşme ağı içerisinde bilginin sağlıklı iletiminde güvenlik problemini doğurmuştur. Günümüz teknolojisinde bilgi güvenliğinin esas olduğu düşünüldüğünde çoğu gizli veya özel olan bu bilgiler, gerekli korumanın sağlanması amacıyla güvenlik mekanizmasına ihtiyaç duyarlar. Bu nedenle sayısal verilerin saklanmasında ve iletiminde bilgi güvenliği, çözülmesi gereken önemli bir konudur.

Bilgi güvenliği temel olarak gizlilik, bütünlük ve erişilebilirlik kavramlarıyla ilgilidir. Bilginin, yetkisiz kişi, kuruluş veya bir işlem tarafından açığa çıkarılamama özelliği gizlilik; doğruluğunu ve eksiksizliğini koruma özelliği bütünlük; yetkili birim tarafından sürekli kullanılabilir özelliği ise erişilebilirlik kavramıyla açıklanır. Bilgi güvenliği için bu özelliklerin korunması ve sağlanması gerekir. Aksi takdirde aktarılan ve işlenen bilgilerin gizliliği ve bütünlüğü saldırı tehditlerine karşı savunmasız kalacaktır. Örneğin, Kevin Poulsen adlı bilgisayar korsanı bir radyo istasyonunun düzenlediği yarışmada telefon hattına sızarak kendisini 102. arayan kişi olarak göstermiş ve Porche marka araba kazanmıştır. Bir başka siber güvenlik olayında, ShadowCrew adlı bir grubun lideri olan Albert Gonzalez, iki sende 17 milyon kredi kartı ve kart numarası ele geçirmiş ve ABD nüfusunun büyük bir kısmının kişisel bilgilerine ulaşarak bu bilgileri para karşılığında satmıştır [2]. Bu somut örneklerdeki saldırılar kişisel menfaate yönelik olmasına karşın bilişim ortamındaki saldırılar genellikle farklı amaçlara hizmet etmektedir.

Farklı haberleşme ağları üzerinden yaygın iletişim, multimedya verilerinin güvenliğini zorlamakla beraber bu ciddi problem aynı zamanda araştırmacılar için de ilgi çekici bir alan olmuştur. Bu durum, araştırmacıları bilinen tekniklerin dışında farklı güvenlik önlemleri almaya yöneltmiştir. Günümüzde sayısal sistemlerde saklanan veya

bilgisayar ağları üzerinden gönderilen bilgilerin gizliliğini ve güvenliğini sağlamada klasik yöntem olarak şifreleme kullanılmaktadır. Şifreleme, içeriği sadece yasal alıcısı tarafından elde edilebilen bir yöntemdir. Yetkisiz dinlemelere karşı multimedya verilerinin korunmasına yönelik olarak kriptografi, steganografi gibi teknikler vardır. Bunlar arasında kriptografi yüksek güvenlik seviyesi sağlamasıyla önemli tekniklerden biri haline gelmiştir. Kriptografi, bilginin anlaşılabilir ve anlaşılamaz formları arasında dönüştürme tekniklerinin geliştirilmesiyle ilgilenir. Bu teknik, gizlilik ve erişim kontrolünü içerir [3] ve bu sayede bilginin güvenli olmayan kanallar üzerinden gönderilmesi veya sayısal ortamlarda saklanması mümkün olur [4].

Şifreleme yöntemi dışında sayısal verilere eklenen sinyal niteliğinde olan sayısal filigranlama teknolojisi de multimedya veri güvenliği için geliştirilmiştir. Bu teknik bilginin sayısal multimedya içeriğine gömülme işlemidir ve bunun sonucunda bilgi, yasadışı kopyalama ve değişimine karşı korunabilir hale gelmektedir. Bu teknoloji kontrol ve kimlik doğrulaması gibi güvenliğe dayalı farklı amaçlar için kullanılabilir [5]. Multimedya veri güvenliğini sağlamada şifreleme ve sayısal filigranlama tekniklerinin birlikte kullanıldığı çalışmalar da bu alanda mevcuttur [6].

Multimedya bilgileri arasında özellikle sayısal görüntüler, sosyal toplumdaki gerçek zamanlı görsel haberleşmenin yaygınlaşmasıyla beraber insanların günlük yaşantılarında önemli bir rol oynamaktadırlar. Bu anlamda, görsel bilginin korunumu son derece önemlidir. Görüntü güvenliği protokolü, iletilen bilginin açığa çıkmasına ve iletim sırasında değişmesine karşı veriyi koruyan uygulama katman teknolojisidir. Sayısal görüntü şifreleme, görüntü işlemenin en önemli yöntemlerinden birisidir. Görüntü şifreleme teknikleri ağırlıklı olarak sıkıştırma yöntemini, kriptografi mekanizması ve kaos teknikleri içermektedir [7]. Günümüzde gelişmiş şifreleme standardı (AES), veri şifreleme standardı (DES), uluslararası veri şifreleme algoritması (IDEA) ve RSA (Ron Rivest, Adi Shamir, Leonard Adleman) tekniği gibi farklı kriptografik algoritmalar veri şifreleme amacına yönelik olarak ortaya atılmıştır. Genel olarak multimedya verisi durgun yani gerçek zamanlı bir akışın olmadığı formatta ise bu bilgi sıradan ikili veri olarak düşünülebilir ve burada bilinen şifreleme teknikleri kullanılabilir. Ancak genellikle görüntü bilgisi, metin bilgisinden çok daha büyük olduğundan geleneksel kriptografik sistemlerle görüntü bilgilerini doğrudan şifrelemek çok fazla zaman gerektirmektedir. Dolayısıyla, evrensel şifreleme yapıları, görüntü ölçüsü büyük olduğu zaman düşük seviyeli verimliliğe sahiptirler [3,4]. Ayrıca, AES veya IDEA gibi şifreleme teknikleri genellikle metin tabanlı

bilgilerin şifrelenmesinde kullanılmaktadırlar. Görüntü bilgileri, sahip oldukları büyük veri kapasitesi, yüksek piksel artıklığı ve yakın pikseller arasındaki güçlü bağıntı özellikleri ile metin bilgilerinden çok farklı yapıdadırlar. Bu içsel özelliklerinden dolayı görüntü bilgilerini, mevcut şifreleme algoritmalarıyla şifrelemek uygun bir çözüm değildir. Bunun başlıca nedeni, etkili bir görüntü şifreleme algoritması tasarımında kaynak görüntünün bilinen kriptografik yöntemlerle hızlı bir şekilde karıştırılması ve dağıtılmasının çok zor olmasıdır [8]. Ayrıca görüntü bilgisine yönelik şifreleme işlemi, çok fazla işlem gücü ve bant genişliği gerektirmekte bunun sonucunda da verimsiz bir şifreleme gerçekleşmektedir [9-12]. Bu algoritmalar ile karşılaştırıldığında, kaos tabanlı algoritmalar karmaşıklık, hız, işlem gücü ve güvenlik açısından daha üstün bir performans göstermiştir [13].

Görüntü bilgisine ait şifre çözme yapısı bir başka yönü ile de metin tabanlı şifreleme yapısından farklıdır. Şifrelenmiş metin bilgisinin çözümlenmesinde elde edilen bilginin boyutu, orijinal metin boyutu ile aynı olmalıdır. Ancak bu gereklilik, görüntü bilgisi için zorunlu değildir çünkü insan algısının karakteristiği çözülmüş görüntüdeki küçük bir bozulmayı ayırt edemeyecek seviyededir. Bu durum, bilinen şifreleme algoritmalarının görüntü şifrelemede tercih edilemeyeceğinin bir nedeni olarak da gösterilebilir [3].

Sayısal görüntülerin yasadışı yollarla çoğaltılmasını önlemek ve üzerinde izinsiz değişiklikler yapan yetkisiz kullanıcılardan korumak amacıyla farklı görüntü şifreleme yapıları sunulmuştur. Mevcut görüntü şifreleme teknikleri kullanan bu fikirler üç ana kategoride sınıflandırılabilir. Görüntüye yönelik piksel pozisyon permütasyonu, piksel değer dönüşümü ve bunların kombinasyonu olan yapı bu teknikleri oluşturur [14-17]. Pozisyon permütasyonu, görüntü içerisindeki piksel pozisyonlarının karıştırılmasına yönelik bir işlemdir ve genellikle düşük güvenliğe sahiptir. Değer dönüşümü algoritması ise orijinal görüntü bilgisindeki piksel değerinin değiştirilmesine dayalı olup, düşük hesaplama karmaşıklığına ve basit donanım potansiyeline sahiptir. Son olarak bu her iki yapının da kullanıldığı bütünleşik yapıda piksellere ait hem pozisyon hem de değer değişimi gerçekleştirilir ve genellikle yüksek güvenliğe sahiptir [17].

1.2. Tezin Amacı

Bu tezde, sayısal görüntüler için kaos tabanlı simetrik bir kripto sistem tasarımına yönelik bir çalışma yapılmıştır. Sosyal toplumdaki gerçek zamanlı görsel haberleşmenin yaygınlaşması sonucu sıkça kullanılan kişisel görüntüler, insanların günlük yaşantılarında önemli bir rol oynamaktadır. Dolayısıyla, bu verilerin saklanmasında ve iletiminde

güvenliğin üst seviyede olması esastır. Görüntü bilgilerini, sahip oldukları içsel yapı itibariyle geleneksel şifreleme teknikleri kullanarak şifrelemek verimsizdir. Çünkü bu işlem, büyük bant genişliği ve çok fazla işlem gücü (kaynak tüketimi) gerektirmektedir.

Sunulan görüntü kripto sisteminde hızlı, güvenli ve verimli bir şifreleme amaçlanmıştır. Ayrıca şifreleme ve çözme algoritmalarının donanım ortamında gerçekleştirilmesi amaçlanarak uygulama noktasında kullanılabilirliği sorgulanmıştır.

1.3. Tezin İçeriği

Birinci bölümde teze giriş yapılmış, mevcut ve önceki çalışmalar ele alınmış ve tezin amacı anlatılmıştır.

İkinci bölümde kriptoloji ile ilgili temel kavramlardan bahsedilmiş ve farklı şifreleme yapılarına yer verilmiştir. Ayrıca, Vernam algoritmasında kullanılan tekniğin önemine değinilerek güvenli bir şifreleme için gerekli koşullar açıklanmıştır.

Üçüncü bölümde kaos kavramının kriptografi içerisindeki yeri ve ortak noktaları ele alınmıştır. Burada, kaos tabanlı kripto sistemler ile ilgili bilgilere yer verilmiş ve bu alanda gerçekleştirilmiş bilimsel çalışmalara değinilmiştir. Ayrıca tasarlanacak olan görüntü şifreleme algoritmasında kullanılacak yöntem ve metotlara yer verilmiştir.

Dördüncü bölümde standart Lojistik harita modelinin denklem sistemi kullanılarak iyileştirilmiş yeni bir kaotik harita detaylı bir şekilde anlatılmış ve bu sistemin kriptografik anlamda kod üreteci olarak kullanılmasına yönelik tüm istatistiksel analizleri incelenmiştir.

Beşinci bölümde sunulan kripto sistemin tasarımına yönelik tüm detaylar anlatılmış ve şifrelemede kullanılacak olan gizli anahtar kodları için önemli analizler yapılmıştır.

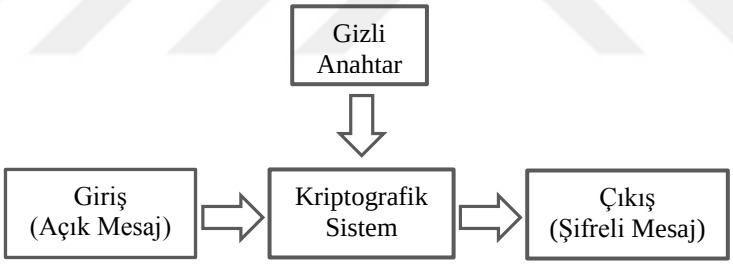
Altıncı bölümde sunulan kripto sistemin performans ve güvenlik analizleri tüm hatlarıyla incelenmiştir. Ayrıca kripto sistemde kullanılan şifreleme algoritmasının istatistiksel ve diferansiyel analizlerine yer verilmiş ve bu sonuçlar mevcut bazı şifreleme algoritmalarıyla karşılaştırılmıştır.

Yedinci bölümde tasarlanan görüntü kripto sisteminin uygulama noktasında donanım ortamında gerçekleştirilmesi ele alınmış ve tüm sistemin uygulanabilirliliğine yönelik sonuçlarına yer verilmiştir.

Sekizinci bölümde ise tezde uygulanan yöntemlerden elde edilen teorik ve deneysel sonuçlara yer verilmiş ve ileriye dönük çalışmalar için önerilerde bulunulmuştur.

2. KRİPTOLOJİ

Şifreleme teknolojisi olarak adlandırılan kriptoloji, güvenli haberleşme teorisi ile uğraşan bir bilim dalıdır [4]. Ulusal güvenlikte, askeri, diplomasi alanlarında ve bilgisayar verilerinin korunmasında kriptoloji sıkça kullanılmaktadır. Haberleşme ve bilgi teorisine, kaos teorisine, bilgisayar, kripto analiz, kriptografi ve matematik bilimlerine de katkı sağlayan kriptoloji, kriptografi ve kripto analiz olmak üzere iki alt dala ayrılır [18]. Kriptografi, hızlı ve güvenli bir şifreleme algoritmasının nasıl tasarlanacağını incelerken; kripto analiz ise mevcut algoritmaların güvenlik açığını bulmaya çalışır ve bunların bazı saldırılara karşı savunmasız olup olmadığını araştırır. Bilgi mesajının öngörülemeyen şekilde okunamayan bir formata dönüştürerek bilgiyi gizli tutmak, kriptografik bir sistemin temel amacıdır. Bu işlem için öncelikli olarak bir şifreleme düzenine ihtiyaç vardır. Bu düzen şifreleyici ya da kripto sistem olarak adlandırılır ve Şekil 2.1' de gösterilen şematik bir yapıya sahiptir [5].



Şekil 2.1 Genel bir kripto sistemin yapısı

Kripto sistem, bilgi güvenliğinin sağlanmasına yönelik uygulanan kriptografik tekniklerin ve beraberindeki altyapılarının gerçekleştirilmesidir. Böyle bir kripto sistemde işlem sonunda hedeflenen amaç bilgi mesajının sadece alıcısı tarafından çözümlenmesidir. Bir kripto sistemin bileşenleri şu şekilde özetlenebilir:

- Açık bilgi mesajı (plain-text): İletimi sırasında korunması gereken veridir.
- Şifreleme algoritması: Açık bilgi mesajı ile şifreleme anahtarını kullanarak şifreli bilgi üreten matematiksel işlemdir.

- Şifreli mesaj (cipher-text): Belirli bir şifreleme anahtarı kullanılarak şifreleme algoritması tarafından oluşturulmuş okunamayan bilgidir veya kısaca bilgi mesajının karıştırılmış halidir.
- Çözme algoritması: Şifreli bilgi mesajından çözme anahtarı kullanılarak orijinal bilginin elde edildiği matematiksel işlemdir. Şifreleme işleminin tam tersidir.
- Şifreleme anahtarı: Kriptografik bir algoritmada şifreli bilgi üretmek için kullanılan değişken değerlere sahip gizli kodlardır.
- Çözme anahtarı: Alıcısı tarafından bilinen ve çözme algoritmasında orijinal bilginin tekrar elde edilmesinde gerekli olan gizli kodlardır. Çözme anahtarı, şifreleme anahtarı ile aynı olmak zorunda değildir.

Kriptografi temelde bilginin karıştırılmasını sağlayarak verilerin doğruluğunu veya güvenliğini garanti altına alır ve böylelikle izinli alıcı dışında hiç kimse tarafından bu bilgi okunamaz. Diğer taraftan saldırgan, bilgi mesajını belirlemeye çalışan izinsiz varlıktır. Bu girişim, şifrelenmiş mesaja erişebilir hatta çözme algoritmasına ulaşabilir. Ancak çözme anahtarının saldırgan tarafından kesinlikle bilinmemesi gerekir. Bir sistemin güvenliği algoritmanın gizli tutulmasına bağlı değildir. Kriptografinin temel ilkesi sistem güvenliğinin sadece anahtara bağlı olmasını gerektirir. Bu ilke, Kerckhoff prensibi olarak bilinir [19]. Dolayısıyla kripto sistemin tasarımı, sistem güvenliğinde önemli bir rol almaz. Modern kriptografide önemli olan şifrelenmiş bilgi güvenliğinin sadece şifreleme anahtarının güvenliğine bağlı olmasıdır. Bu özellik, Shannon özdeyişi olarak bilinen "düşman, kullanılan sistemi bilir" ifadesi ile de desteklenmiştir [20, 21].

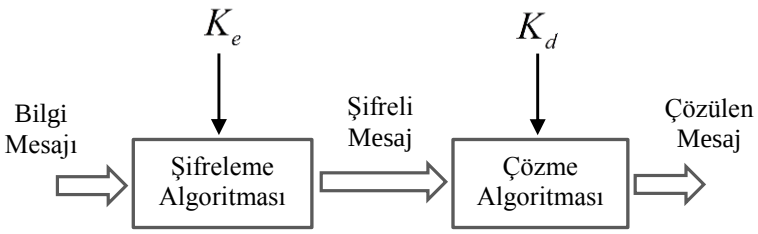
Belirli bir kripto sistem için olası tüm çözme anahtarlarının permütasyon kümesi anahtar alanı olarak adlandırılmaktadır. Kaba-kuvvet (Brute-force) saldırısı kullanan bir saldırganın şifreleme anahtarını ortaya çıkarmak adına kullandığı kapsamlı anahtar tarama yöntemini uygulanamaz hale getirmek için anahtar alanı yeterince büyük olan bir kripto sistem tasarlanmalıdır. Kaba-kuvvet saldırı tekniği, bir şifreleme algoritmasına yönelik olarak gerçekleştirilen en etkili şifre kırma yöntemlerinden birisidir. Bu teknikte, saldırgan kripto sistemin olası tüm anahtar alanı içerisinde tam bir tarama gerçekleştirerek doğru anahtarı bulmaya çalışır [22]. Örneğin, 8-bit uzunluğunda şifreleme anahtarının kullanıldığı bir kripto sistemde anahtar alanı $2^8 = 256$ olası anahtar içerir. ABD Ulusal Standart ve Teknoloji Enstitüsü tarafından kabul edilmiş ve DES algoritması yerine geliştirilen AES, 256-bitlik simetrik anahtar kullanmakta ve bu da olası 1.15×10^{77} sayıda muhtemel anahtar içermektedir. Bu seviyedeki bir işlemin kaba-kuvvet saldırısıyla

kırılabilmesi için evrenin yaşından çok daha uzun bir süre gerekmektedir. Pratiksel olarak bunun uygulanamaz olduğu açıktır. Sonuç olarak, kripto sisteme ait anahtar alanı yeterince büyük olduğunda bu saldırı tekniği başarısız olur. Yüksek seviye güvenlik için anahtar alanı ölçüsü 2^{100} değerinden büyük olması gerekir. Bu seviye sağlandığında kaba-kuvvet saldırısı teknik anlamda başarısız olacaktır [23,24].

Anahtar aramaya dayalı kaba-kuvvet saldırısı dışında kripto analiz ve sistem tabanlı saldırı yöntemleri, şifrelenmiş bilgilere karşı yapılan saldırı tekniklerindendir. Bir kripto sistemde doğrudan kriptografik algoritmaya yönelik gerçekleştirilmeyen izinsiz müdahale sistem tabanlı saldırılar altında incelenir. Kripto analiz tekniğinin amacı ise şifrelemedeki algoritmayı kırmak ve burada kullanılan anahtarı veya açık bilgiyi ortaya çıkarmaktır. Çoğu şifreleme algoritması, gelişmiş matematiksel kombinasyon yöntemleri ve hesaplama gücü ile kırılabilir. Aynı zamanda şifrelenmiş birçok bilginin, şifreleme anahtarı bilinmeden çözülebildiğini gösteren çalışmalar da vardır. Bir kripto sisteme ait güvenlik ölçüsü şu şekilde özetlenebilir: Kripto sistemden üretilen şifrelenmiş bilgi ile ilgili ne kadar çok veri olursa olsun ya da saldırgan bu bilgiyi kırmak adına ne kadar çok hesaplama gücüne sahip olursa olsun bu şifreli bilgi, buna karşılık gelen açık bilgiyi tek başına belirlemekte yeterli bilgi içermiyorsa bu şifreleme algoritması koşulsuz olarak güvenlidir. Bir şifreleme algoritmasından elde edilen şifrelenmiş bilginin kırılması için kullanılan maliyet, kaynak bilginin değerini aşıyor ya da şifreyi kırmak için gereken süre bilginin kullanım ömründen fazla ise bu şifreleme algoritması hesaplama açısından (şartlı olarak) güvenlidir [25]. Günümüzde kullanılan AES, RSA gibi çoğu şifreleme algoritması hesaplama açısından güvenlidir.

2.1. Simetrik ve Asimetrik Kripto Sistemler

Bir kripto sistemde anahtar üretimi ile birlikte şifreleme ve çözme algoritma yapıları simetrik ve asimetrik kriptografi başlığı altında incelenmektedir. Simetrik ve asimetrik algoritmalar için öncelikle kripto sistem yapısı detaylı bir şekilde ele alınması gerekir. Bu anlamda, genel bir kripto sisteme ait şifreleme ve çözme yapısı Şekil 2.2' de gösterilmiştir.



Şekil 2.2 Kripto sistemlerin şifreleme ve çözme yapısı

Kripto sistemlerin şifreleme prosedürü, Denklem (2.1)' de verildiği gibi basit bir matematiksel ifade ile temsil edilebilir.

$$C = E\left[K_e(P)\right] \tag{2.1}$$

Burada *P*, bilgi mesajını gösterirken; *C* ise şifreli mesajı ifade eder. Şifreleme işlemi *E(.)* fonksiyonu ile tanımlanırken; K_e ise şifreleme anahtarını temsil eder. Benzer şekilde Denklem (2.2)' de belirtilen çözme işlemi,

$$P = D\left[K_d(C)\right] \tag{2.2}$$

gibidir ve burada K_d, çözme anahtarını; *D(.)* ise çözme fonksiyonunu gösterir. Eğer $K_e = K_d$ ise kripto sistem, simetrik şifreleyici veya gizli anahtar şifreleyici olarak adlandırılır. Simetrik şifrelemede, şifreleme anahtarları göndericiden alıcıya gizli ve farklı bir kanal üzerinden aktarılır. Buna karşın $K_e \neq K_d$ şeklinde tasarlanan bir kripto sistem ise asimetrik (genel anahtar) şifreleyici olarak adlandırılır [26]. Burada, şifreleme anahtarı K_e herkese sunulurken, çözme anahtarı olan K_d ise iletilmez gizli tutulur ve anahtar transferi için fazladan bir gizli kanala gerek duyulmaz. Dolayısıyla, asimetrik algoritmada herhangi bir kimse mesaj şifreleyebilir ancak sadece doğru ve kişisel anahtarı olan kişi bu mesajı çözebilir. Böyle bir kripto sistem için şifreleme anahtarından çözme anahtarının kabul edilebilir bir zaman içerisinde elde edilmesinin mümkün olmadığı kabul edilir [5]. Simetrik şifreleme, genellikle bilgiyi korumada kullanılırken; asimetrik şifreleme ise kullanıcı veya bilgisayarın kimliğini doğrulamak ya da bilginin güvenilir olduğunu kanıtlamak için geliştirilmiştir. Asimetrik algoritmalar çok daha fazla matematiksel hesaplama

gerektirdiğinden bunlar simetrik algoritmalara göre daha yavaştırlar [27]. AES ve DES simetrik şifrelemeye örnek olurken, RSA asimetrik şifrelemenin kullanıldığı bir algoritmadır.

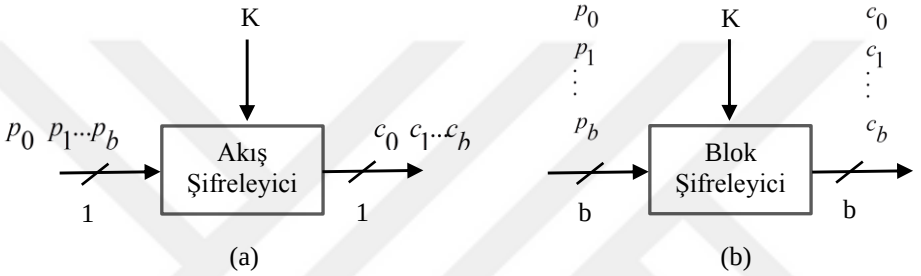
Simetrik kripto sistemler verinin şifrelenmesinde veya çözümlenmesinde sadece bir anahtar kullanır. Her iki işlemde kullanılan anahtar, özel veya gizli anahtar olarak tanımlanır ve yalnızca şifreleme/çözme işlemlerinde yetkisi olan kişi ya da sistem tarafından bilinir. Pratik uygulamalarda kullanılan bu anahtar, taraflar arasındaki gizliliği sağlamak için paylaşılan ortak sırrı temsil eder. Dolayısıyla her iki tarafın da gizli anahtara erişimi olması gerekliliği simetrik şifrelemenin en büyük dezavantajıdır. Anahtar paylaşımı için ortak bir ağın kullanıldığı sistemlerde simetrik şifreleme tercih edilmez. Çünkü akıllı bir saldırgan böyle bir ağdaki trafik akışını analiz ederek bilgi mesajını çözebilir ve hatta orijinal yerine gerçek görünen sahte bilgiyi ilave edebilir. Dolayısıyla iletilen bilgiye yönelik kötü niyetli ekleme ve değiştirme olasılığı simetrik anahtar şifrelemede daha yüksektir. Buna karşın simetrik şifreleme, asimetrik olanlarına göre daha az karmaşıktır.

Asimetrik kripto sistemlerde gönderici ve alıcı için birinin diğerinden türetildiği iki farklı anahtar kullanılır. Dolayısıyla haberleşme anlamında gönderici ve alıcı ortak bir gizliliğe gerek duymaz. Ancak burada kullanılan algoritma, matematiksel olarak karmaşık hesaplamalar içerdiğinden asimetrik kripto sistemler, simetrik olanlara göre daha yavaş ve daha düşük verimliliğe sahiptirler. Diğer taraftan, simetrik şifreleyiciler ise yüksek veri şifreleme oranına sahip olacak şekilde tasarlanabilirler. Sonuçta çoğu uygulamada her ikisinin bir kombinasyonu kullanılmakta ve bütünleşik olan bu yapı hibrit şifreleme olarak bilinmektedir [28]. Burada kullanılan genel ve özel anahtarlar güvende olduğu sürece bu algoritma, yüksek seviye güvenlik sağlayan bir şifreleme türü olarak kabul edilmektedir.

2.2. Akış ve Blok Şifreleyiciler

Simetrik şifreleyiciler yapısına göre akış şifreleyici (Stream cipher) ve blok şifreleyici (Block cipher) olarak sınıflandırılır [4]. Blok şifreleme, kaynak bilgiye ait bir bloğun, bir bütün olarak ele alındığı ve üretilen çıkışın blok uzunluğu ile kaynak bilginin blok uzunluğunun aynı olduğu bir şifreleme yapısıdır. Örneğin, bir blok şifreleme algoritması giriş olarak 128-bit uzunluğunda bilgi mesajını alıp, çıkış olarak buna karşılık gelen 128-bit uzunluğundaki şifrelenmiş bloğu oluşturabilir. Blok şifreleyici esasında şifrelenen ve çözülen tüm bloklarda aynı anahtarların kullanıldığı bir simetrik anahtar şifreleyicidir. Böyle bir algoritmaya ait daha üst seviye güvenlik için blok uzunluğunun ve

anahtar boyutunun daha büyük tutulması gerekir. Akış şifreleyici ise bilgiye ait veri akışında bir bit veya bir bayt verinin tek seferde şifrelendiği algoritma yapısıdır [5]. Akış şifreleyiciler bilgiyi, tek bir basit bit gibi küçük bloklar halinde işler ve şifreleme işlemi bu bilgi işlendikçe değişebilir dolayısıyla böyle bir algoritma kullanan kripto sistemin bir bellek yapısına sahip olduğu söylenebilir. Burada şifreleme işlemi, sadece anahtar ve bilgiye bağlı olmadığı gibi aynı zamanda mevcut duruma da bağlılık gösterebilir [29]. Akış şifreleyiciler donanım ortamında genellikle blok şifreleyicilere göre daha hızlıdır ve daha az karmaşık donanım devresine sahiptirler [29].



Şekil 2.3 Şifreleme ilkesi; a) Akış şifreleyici, b) Blok şifreleyici

Burada bilgi mesajı, anahtar ve şifrelenmiş bilgi, $p_i, k_i, c_i \in \{0,1\}$ gibi bit değerlerinden oluşur. Anahtar akışının kaynak bilgi veya şifrelenmiş bilgiden bağımsız olarak üretildiği yapı eş zamanlı akış şifreleyici; bu yapının şifreleme anahtarı dışında önceki şifreli karakterlere olan bağlılığı da eş zamansız akış şifreleyici olarak adlandırılmaktadır [29]. Çoğu pratik akış şifreleyici, eş zamanlı yapıda olup, bu şifreleyiciler küçük ve hızlı olma eğilimine sahip olduklarından dolayı büyük hesaplama gerektirmeyen uygulamalar için daha uygundurlar. Akış şifrelemede bit değerleri ayrı ayrı şifrelenir ve bu işlem anahtar akışından tek bit değerinin sırayla bilgi mesajındaki bir bit değerine eklenmesiyle gerçekleştirilir. Burada basitçe gerçekleştirilen aritmetik modül-2 toplama veya mantıksal fonksiyon olarak özel-veya (XOR) operasyonudur ve matematiksel olarak Denklem (2.3)' deki gibi ifade edilir.

$$C_i = E_k(P_i) \equiv \left(k_i + p_i\right) \bmod 2 \tag{2.3}$$

XOR işlemi modern kriptografide önemli bir rol oynamaktadır ve bu tezin geri kalanında birçok kez kullanılacaktır. Bu işleme ait doğruluk tablosu, Tablo 2.1' de gösterilmiştir.

Tablo 2.1 XOR fonksiyonunun davranışı

p_i	k_i	$c_i = (k_i + p_i) \bmod 2$
0	0	0
0	1	1
1	0	1
1	1	0

XOR fonksiyonunun kriptografide şifreleme işleminin temelinde kullanılmasının birkaç nedeni bulunmaktadır. Bu nedenler arasında en önemlisi bu fonksiyonun mükemmel bir şekilde dengeli davranmasıdır. Burada dengeli davranmak, anahtar değerine bağlı olarak şifrelenmiş bit değerinin tam olarak ½ olasılıkla '0' veya '1' olabilmesidir. Örneğin, şifrelenecek olan bilgi mesajına ait bit değeri $p_i = 1$ olması durumunda, %50 olasılıkla şifrelenmiş bit bilgisi '1' veya '0' dır. Aynı durum $p_i = 0$ için de geçerlidir. Dolayısıyla şifrelenmiş bit bilgisi eğer '0' ise, bilgi biti '0' olabildiği gibi '1' de olabilir. Sonuçta, şifrelemede kullanılacak anahtar biti mükemmel bir şekilde rastgelelik içeriyorsa yani tam olarak %50 olasılıkla '0' veya '1' değerlerinden oluşuyorsa, muhtemel her iki şifreli bit bilgisi de çıkışta %50 olasılıkla oluşacaktır. Bu özellik XOR fonksiyonunu, diğer lojik fonksiyonlardan ayırır [30]. Ayrıca XOR, basit ve hızlı çalıştırılabilen bir fonksiyon olmasının yanında tersi de alınabilen bir işlemdir. Bu sayede şifrelenmiş veri üzerinde aynı fonksiyon gerçekleştirilerek tekrar bilgi mesajına ulaşmak mümkündür. Bu özellik Denklem (2.4)' deki gibi gösterilebilir.

$$P_i = D_k(C_i) \equiv (k_i + c_i) \bmod 2 \tag{2.4}$$

Denklem (2.3)' de verilen şifreleme algoritmasında bilgi mesajının elde edilmesinde gerekli çözme yapısı Denklem (2.4)' de tanımlanmıştır. Burada şifreleme denklemi, Denklem (2.5)' deki çözme fonksiyonu içerisinde kullanıldığında,

$$D_k(c_i) = (k_i + c_i) \bmod 2 = (k_i + p_i + k_i) \bmod 2 = (2k_i + p_i) \bmod 2 = p_i \tag{2.5}$$

şifrelenmiş bilgi üzerinde aynı işlemin kullanılmasıyla orijinal mesajın tekrar elde edilebildiği görülmektedir. Böyle bir kodlamanın gerçekleştirildiği sistemde kaynak bilginin bit değerleri rastgele bir şekilde değiştirilerek bilgi saklanmaktadır. Dolayısıyla, şifreleme anahtarını bilmeyen izinsiz bir kullanıcı kaynak bilgide hangi bitin değiştiğini (bit akışında '1' oluşumuna karşılık gelen) veya hangisinin değişmediğini (bit akışında '0' oluşumuna karşılık gelen) bilemeyecektir [4]. Bilgi mesajına rastgelelik olgusunun eklenmesiyle şifreleme işlemi olasılık içeren bir duruma dönüşerek, bilgilerin kendi içerisinde ilişkilendirilmesi engellenir.

2.3. Vernam Algoritması

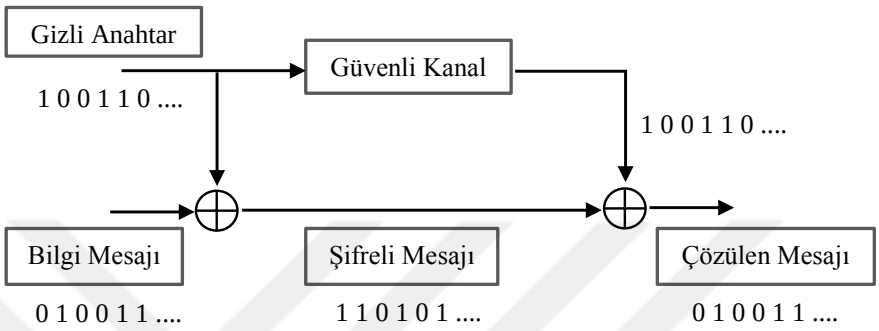
Basit bir XOR işlemi ile gerçekleştirilen şifreleme yapısı, bilinen algoritmalar arasında matematiksel olarak kırılamayan tek güvenli yöntem olan Vernam şifreleme algoritmasında da kullanılmıştır. Tek kullanımlı anahtar (One-time pad, OTP) olarak da bilinen Vernam kripto algoritması, 1917 yılında Gilbert Vernam tarafından uzak yazıcı uygulaması için geliştirilmiştir. Bu sistem, mükemmel bir gizliliğe sahiptir yani şifrelenmiş veri kesinlikle kaynak bilgiyle alakalı hiçbir ek bilgi vermez. Burada tek kullanımlı kavramı, şifrelemede kullanılan anahtarın sadece bir kez kullanılabilir olmasından gelmektedir [4, 31].

Bir kripto sistemin mevcut hesaplama teknolojilerin ve algoritmaların kullanılarak makul bir süre içerisinde şifreleme anahtarını kırma olasılığı son derece küçük ise bu kripto sistem güvenli olarak kabul edilir. Tek kullanımlı şifreleme dışındaki tüm kriptografik algoritmalar yeterli zaman ve sayıda şifreli çıktı verildiğinde teorik olarak kırılabilir. Tek kullanımlı şifrelemede kullanılan anahtarlar, dört önemli kuralı takip eder [5].

- Anahtar, şifrelenecek veri veya mesaj uzunluğunda olmalıdır.
- Anahtar, tamamen rastgele olmalıdır. Basit bir bilgisayar fonksiyonundan ya da benzeri bir uygulamadan üretilmemelidir.
- Her bir anahtar sadece bir defalık kullanılmalıdır. Hem gönderici hem de alıcı anahtarları kullandıktan sonra yok etmelidir.
- Biri alıcı diğeri de vericide olmak üzere anahtarın sadece iki kopyası olmalıdır.

Bu kurallar doğru ve eksiksiz bir şekilde uygulandıktan sonra tek kullanımlı şifreleme algoritması, sonsuz hesaplama gücü veya sonsuz zaman altında bile kırılamaz olduğu kabul edilir. Çünkü matematiksel olarak bu algoritmanın kırılması mümkün değildir. Anahtarlar

tek olduğu ve yeniden kullanılmadığı müddetçe hiçbir istatistiksel analiz veya örnek eşleştirme teknikleri kripto analizciler tarafından uygulanamaz. Ayrıca kullanılan anahtar akışı tamamen rastgele ise sistem çıkışında oluşan şifrelenmiş bilgi de tamamen rastgele olacaktır. Bu durum, şifrelenmiş bilgi ile orijinal bilgi arasında hiçbir ilişkinin olamayacağını ifade eder. Vernam algoritmasına ait şema Şekil 2.4' de gösterilmiştir.



Şekil 2.4 Vernam algoritmasının blok şeması

Simetrik Vernam sisteminin en önemli hassas görünüşü sahip olduğu anahtar serisinin gerçek rastgelelik içermesidir. Bir durum serisi için bir sonraki durumun tahmini, bu işlemi üreten tüm parametrelerin bilinmesine rağmen olanaksız ise böyle bir serinin tamamen rastgele olduğu söylenebilir. Bilgisayarda çalışan bir yazılım gibi herhangi bir deterministik sistem hiçbir şekilde gerçek rastgele sayılar üretemez. Çünkü bilgisayar sistemindeki bir sonraki durum, makinanın mevcut durumundan tamamen belirlenebilir. Şifreleme anahtarı için rastgele sayılar, hiçbir zaman sadece yazılım tarafından üretilmemelidir. Bunlar, doğası itibariyle tamamen deterministik olmayan işlemler tarafından oluşturulmalıdır [5].

Shannon, kırılamaz bir şifreleme sisteminin karakteristiğini şu şekilde tanımlamıştır: Şifreleme anahtarı tamamen rastgele, en azından kaynak bilgisi kadar büyük, tamamen ya da kısmen tekrar kullanılmamış ve gizli tutulur olmalıdır. Shannon tarafından sunulan bu kriterlerin, tek kullanımlı anahtar sistemindeki özellikler ile aynı olduğu açıktır. Bruce Schneier, anahtarın tamamen rastgele ve tekrardan kullanılmama şartıyla tek kullanımlı anahtar sistemini "mükemmel şifreleme yapısı" olarak tanımlamaktadır. Bunun yanında gerçek rastgelelik, matematiksel fonksiyonlarla elde edilemeyeceği gibi sadece belirli fiziksel işlemler ile oluşturabileceğini belirtmiştir. Bu açıdan değerlendirildiğinde, matematiksel formüllerin kullanımı bizi sözde-rastgele sayılara götürecektir ki bunlar

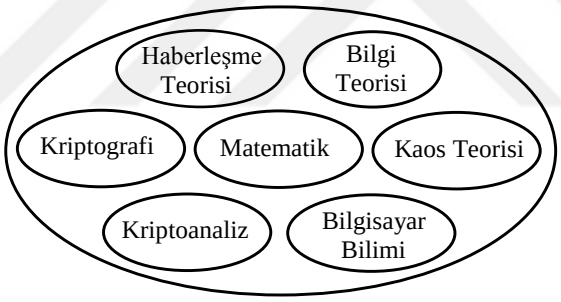
rastgele formda görünmelerine karşın gerçekte önceden belirlenebilir. Buna karşın gerçek rastgele sayılar, atmosferik gürültü veya radyoaktif kaynak gibi fiziksel olayların ölçümlerinden gelmektedirler [5, 32].

Bir kripto sistemdeki güvenlik, kullanılan anahtarın rastgelelik durumuna bağlıdır. Kriptografik bağlamda ikili seri şeklinde oluşan bir şifreleme anahtarında iki temel karakteristik özellik aranır: tahmin edilemezlik ve eşit dağılım. Bu özellikler sırasıyla şu şekilde açıklanabilir: Bit uzunluğundan bağımsız olarak, anahtar akışı içerisindeki herhangi bir değerden sonraki biti tahmin etme olasılığı ½ den daha iyi olamaz ve böyle bir anahtar serisi içindeki '1' ve '0' sayıları birbirine eşit olmalıdır. Bir anahtar üretecinin çıkışı tahmin edilemediği müddetçe kriptografik olarak güvenli üreteç gibi değerlendirilir ve bu şekilde kullanılır. Belirli bir durumun saldırgan tarafından tahmin edilmesinin zor olduğunu belirtmek için genellikle 'öngörülemeyen' terimini kullanılır.

Kaotik sinyallerin yüksek seviyede öngörülemeyen ve rastgele görünümlü yapısı mühendislik biliminde yeni uygulamalara öncülük etmektedir. Bu karakteristik yapı, deterministik kaotik sistemlerin en çekici özellikleridir [33]. Kaotik sistemler, başlangıç koşullarına ve kontrol parametrelerine gösterdiği hassas duyarlılık, sözde-rastgelelik (pseudo-randomness) ve farklı sistem parametreleriyle benzer davranış sergilemesi gibi çok önemli özelliklere sahiptirler. Bu özellikler kriptografide Shannon gereksinimleri olarak bilinen karışıklık ve dağılma işlemlerini karşılamakta olup, bilgi şifrelemede kaotik sistemleri popüler hale getirmektedir. Bu açıdan bakıldığında kaos ve kriptografi bazı ortak özelliklere sahiptirler. Bu ortak özellikler arasında en belirgin olanı ise değişkenler ve sistem parametrelerine karşı olan hassasiyetleridir. Shannon bir çalışmasında, iyi bir karıştırmada dönüşüm fonksiyonlarının karmaşık ve tüm değişkenlere karşı duyarlı olması gerektiğini ifade etmiştir. Böylelikle herhangi bir değişkendeki ufak bir değişim, sistem çıkışını önemli ölçüde değiştirecektir. Bu özellik, kaos teorisinin ve modern kriptografinin temelini oluşturmaktadır. Sonuç olarak, bu iki disiplininin birbirlerinden faydalanıldığına inanılır [33]. Bu anlamda, yeni şifreleme algoritmaları kaotik sistemlerden geliştirilebilir.

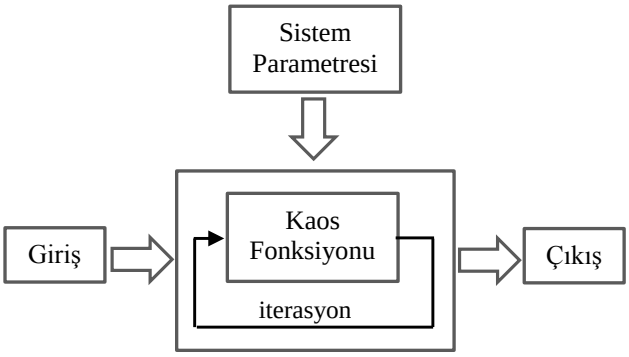
3. KAOS VE KRİPTOGRAFİ

Kaos kelime anlamıyla yunan mitolojisinden gelmiş olup, öngörülemez ya da kestirilemez anlamıyla ilişkili bir kavramdır. Deterministik bir sistemin öngörülemez davranışı kaos olarak da açıklanabilir. Böyle bir sisteme ait davranışın, kolayca tahmin edilebilirliğini düşünmek ilk bakışta doğaldır. Fakat davranışları kararsız ve öngörülemez yapıda olan deterministik sistemler vardır ki bu durum, determinizm tanımının eksikliğinden ziyade tahmin gerektiren dinamiğin çözülemeyen karmaşıklığından gelmektedir. Çözülemeyen bu karmaşıklık, doğrusal olmayan dinamik sistemlerin bir çalışması olan kaos konsepti içerisinde incelenmektedir. Kaos teorisi ise Edward Lorenz tarafından geliştirilmiş, tahmini ve kontrolü mümkün olmayan dinamiklerle ilgilenen matematiksel bir teoridir. Son yirmi yıl içerisinde, kaos teorisinin kullanıldığı bazı uygulama alanları Şekil 3.1' de gösterilmiştir.



Şekil 3.1 Kaos teorisinin uygulama alanları

Kaos teorisi özellikle matematik, fizik, bilgisayar bilimleri ve mühendislik gibi farklı bilimsel alanlarda birçok uygulama bulmuştur. Bu uygulama alanları dışında ayrıca kaotik sinyaller ve sistemler üzerinde gerçekleştirilen başarılı senkronizasyondan sonra bu sistemlerin güvenli ve gizli haberleşmede kullanılmasının önü de açılmıştır [34]. Özellikle son yıllarda kaos tabanlı birçok haberleşme sistemi geliştirilmiş olup, bu tip sistemlerin gerçekleştirilmesinde kullanılan çok sayıda kaotik devre modeli tasarlanmıştır. Kaotik bir sisteme ait devre modeli Şekil 3.2' de gösterilmiştir.



Şekil 3.2 Kaotik haritanın çalışma yapısı

Burada gösterilen kaos fonksiyonu, kaotik harita olarak adlandırılmaktadır ve ayrık zamanlı deterministik bir sistemdir. Bu sistemler tekrarlamalı denkleme sahip olup, matematiksel olarak Denklem (3.1)' deki gibi ifade edilir.

$$x_{n+1} = f\left(x_n\right) \tag{3.1}$$

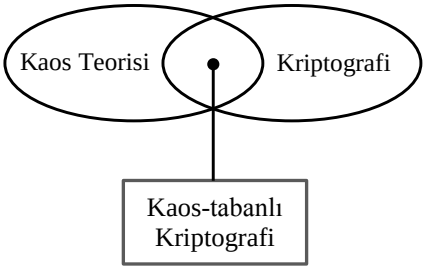
Kaotik bir sistemin başlangıç değeri x_0 ile temsil edilirken; x_n mevcut durumu, x_{n+1} ise bir sonraki çıkışı gösterir. Dikkat edilirse başlangıç durumu, sistemin gelecek durumunu tamamen belirleyecektir. Dolayısıyla aynı giriş bilgileri tamamen aynı çıkış bilgilerini verecektir ki bu durum deterministik sistemin bir sonucudur. Böyle bir sistemden belli bir başlangıç değeriyle oluşturulan sınırlı bir x_i serisinin kaotik olması için asimptotik olarak periyodik olmaması ve en büyük Lyapunov üstelinin pozitif olması gerekir [35]. Kaos varlığının en belirgin özelliği ise dinamik bir sistemde birbirine çok yakın olan başlangıç değerlerinin oluşturduğu yörüngelerin birbirinden üstel olarak ayrılmasıdır. Bu durum, kaos teorisinde kelebek etkisi ile açıklanmaktadır. Bu özellikler kaos kavramının temel karakteristikleridir. Bununla birlikte, doğrusal olmayan tüm dinamik sistemlerin kaos içerdiği de düşünülemez. Ayrıca bir sistem doğrusal ise kesinlikle kaotik olamaz.

Bilgisayar uygulamalarında sonlu bir doğrulukla gerçekleştirilen kaos, kaosun gerçek doğruluğundan ayrılmaktadır. Bu nedenle, bu durum normal olarak "sözde-kaos" (pseudo-chaos) kavramıyla tanımlanır [36]. Dolayısıyla sözde-kaosta, sistem davranışlarında dinamik bozulmalar meydana gelebilir ve yörüngeler birçok anlamda gerçek olanından farklı olabilir. Ancak durum böyle olsa bile, bu sistemlerin dinamik bozulmalarını

minimum seviye çekmek mümkündür. Bu açıdan bakıldığı zaman, yüksek boyutlu kaotik sistemleri kullanma fikri bu anlamda yararlı olabilir.

Kaotik haritalar kararsız dinamik sistemlerdir. Yakınsamalar veya sayısal hesaplama kaynaklı oluşan başlangıç değerlerindeki küçük sapmalar, bu sistemlerde karşılık gelen yörüngelerinde büyük sapmalar yaratarak sistemin uzun dönem tahmin davranışını çok zorlu bir hale getirmektedir. Pratikte belirlenebilir olmayan bu ilkedeki deterministik davranış, entropi üretimi için yerel bir mekanizma şeklinde düşünülebilir. Entropi rastgele bir mesajla ilişkili belirsizlik ölçüsüdür ve mesajın öngörülemezliğini belirler. İşte bu yapı, kaotik haritalarının kriptografide veri şifrelemede kullanılabilirliğini öne çıkarmaktadır. 1949 yılında Shannon' ın kriptografi üzerine sunduğu matematiksel çalışmada kaotik haritalar model olarak ilk defa ortaya atılmıştır. Kaos teorisinin gelişiminden önce sunulan bu çalışmada, kaotik sistemler simetrik anahtar şifreleme mekanizması olarak düşünülmüştür. Kaotik haritaların entropi üretim teorisi, Kolmogorov ve grubu tarafından daha sonra geliştirilmiştir. Shannon, kaotik haritalar kullanarak bilgiye ait ilk verinin ardışık karıştırma yoluyla başarılı bir şekilde şifrelenebildiğini gözlemlemiştir. Böylelikle şifreli veri mümkün olabilecek tüm aralıkta yayılmıştır. Bu şekilde, kripto sisteminin ters dönüşümü bilinmeden, katlanarak karmaşık bir yapıya bürünmüş mesaja ait ilk veriyi elde etmek çok zordur [37].

Kablolu ve kablosuz haberleşme birimlerindeki gelişmeler, son derece güvenli ve hızlı veri şifreleme tekniklerinin geliştirilmesine öncülük etmiştir. Kaotik tabanlı şifreleme sistemleri askeri, mobil iletişim, kişisel veri şifrelemenin yanı sıra akıllı ve güvenli uygulamalar gibi pek çok uygulama alanlarında dikkat çekmiştir. Bu uygulamaların gerçek zamanlı, hızlı ve güvenli olması temel gereksinimlerdir. Büyük hacimde dinamik ve farklı formatta metin, görüntü, ses ve video verilerinin üretildiği bu uygulamalarda güvenlik açısından sinyaller şifreli olarak alıcıya gönderilmelidir. Geleneksel birçok şifreleme algoritması bu amaca yönelik olarak kullanılabilir. Ancak mevcut algoritmalardan bazılarının anlaşılmasının zor, uygulamasının karmaşık ve şifreleme hızlarının yavaş olması, sistem verimliliğini düşürmektedir. Bazıları da gerçek zamanlı uygulamalar için uygun değildir. Dolayısıyla iletişim ağları için son derece güvenli, hızlı ve kolay uygulanabilir şifreleme yapıları sağlayan kaotik sistemler ile ilgili yeni bir yaklaşımın ortaya çıkması hızlanmıştır [38]. Şekil 3.3' de gösterilen bu yaklaşım, veri şifrelemede kaos tabanlı kriptografi kavramını yaratmıştır.



Şekil 3.3 Kaos tabanlı kriptografi şeması

Kaos teorisi odaklı bilgi güvenliği 1992 yıllarında başlamış ve dördüncü nesil gelişimi ile devam etmektedir. Bir ve ikinci nesillerde kaos sadece senkronizasyon işlemlerinde kullanılırken üçüncü ve dördüncü nesillerde ise hem senkronizasyon hem de şifrelemede kullanılmaya başlanmıştır [39]. Kaotik sistemlerin sahip olduğu özelliklerin günümüzdeki kriptografik uygulamalar için mükemmel olduğu kabul edilmektedir. Kaotik şifreleme yaklaşımının başlıca avantajları arasında şifrelemeye ait sistem tasarımında yüksek esneklik ve standart olmayan yaklaşım ile iyi gizlilik söylenebilir. Bunun dışında kaotik sistemlerin basit tasarımı ve bu sistemlerden elde edilen karmaşık ve çok fazla sayıda oluşturulabilecek şifreleme anahtarı, kaliteli anahtar üreteci olmaları açısından bu sistemleri cazip kılmıştır [40]. Shannon ilkelerinden karıştırma ve dağılma özelliklerine benzerlik gösteren başlangıç koşullarına ve kontrol parametresine karşı hassas bağlılık ve öngörülemezlik, kaotik haritaların esasi özellikleridir. Özellikle sistem çıkışlarının rastgele-benzeri olması bu sistemleri, kriptografinin simetrik-anahtar kategorisi için uygun kaynak yapmaktadır. Kriptografi ile kaos dinamiğinin benzer olan yönleri sahip oldukları özellikleriyle birlikte Tablo 3.1' de verilmiştir.

Tablo 3.1 Kaos ve kriptografinin benzer özellikleri

Kaotik Özellik	Kriptografik Özellik	Açıklama
Ergodiklik	Karışıklık	Sistemin çıkışı herhangi bir girdi için benzer dağılıma sahip olması
Başlangıç şartlarına ve kontrol parametresine duyarlılık	Bilgi mesajında/anahtarda küçük bir değişim ile difüzyon	Girişteki küçük bir değişimin, çıkışta büyük bir değişime neden olması
Karıştırma özelliği	Bilgi bloğundaki ufak bir değişikliğin tüm bilgideki difüzyonu	Yerel bir alandaki küçük bir sapmanın tüm alandaki büyük bir değişikliğe sebep olması
Deterministik dinamikleri	Deterministik sözde-rastgelelik	Deterministik bir işlemin rastgele benzeri bir davranış sergilemesi
Yapı karmaşıklığı	Algoritma karmaşıklığı	Basit bir işlemin çok yüksek düzeyde bir karmaşıklığı sahip olması

3.1. Kaos Tabanlı Kripto Sistemler

Kriptografi alanı için kaotik sistemlerin uygulanmasına yönelik analog ve sayısal olmak üzere iki temel yaklaşım izlenmiştir.

3.1.1 Kaos Tabanlı Analog Kripto Sistemler

Kaos tabanlı analog sistemler genellikle güvenli haberleşme alanlarında kullanılmaktadır. Burada açık bilgi mesajı, iletişim kanalının fiziksel seviyesinde kaotik bir sinyal ile gizlenmiştir. Alıcı tarafta ise kaotik senkronizasyon teknikleri kullanılarak gelen sinyal demodüle edilerek bilgi mesajı elde edilir. Bu açıdan değerlendirildiğinde, kaos tabanlı haberleşmenin güvenli iletişimi analog olarak gerçekleştirme potansiyeline sahip olması, bu sistemlerin bilinen haberleşme sistemlerine göre ciddi bir avantajı olarak kabul edilebilir [34].

Kaos tabanlı analog kripto sistemler üç başlık altında sınıflandırılır. Bunlar: kaotik maskeleme (chaotic masking), kaotik anahtarlama (chaotic switching) ve kaotik modülasyondur (chaotic modulation). Son yıllarda bu alanda birçok yeni tasarım önerilmiş olsa da, gerçekte bunların çoğu bu üç temel yapının kısmen değiştirilmiş veya genelleştirilmiş uygulamalarıdır. Bu sistemler tasarımlarının esasi sınırlamalarından dolayı genellikle kriptografik açıdan çok güvenli değildirler. Ancak buna rağmen teknik karmaşıklığı ve yüksek veri iletim hızına sahip olmaları haberleşme alanında bu sistemleri değerli kılmaktadır [41,42].

Kaos tabanlı şifreleme teknikleri, mevcut olan güvenli haberleşme sistemlerine göre yüksek oranda tahmin edilemezlik ve basit uygulama kolaylığına sahiptir. Bu anlamda birçok çalışma haberleşme alanında kullanılmıştır. Örneğin, bir bilimsel çalışmada [43] kaotik analog ve sayısal modülasyon tekniklerinden kaotik maskeleme, kaos kaydırmalı anahtarlama, kaos açma-kapama ve diferansiyel kaos kaydırmalı anahtarlama yapıları için üreteç olarak Lorenz sistemi kullanılmış ve bu tekniklerin Matlab ortamında modellemesi gerçekleştirilmiştir. Bir yüksek lisans çalışmasında ise kaos tabanlı sayısal modülasyon tekniklerinin farklı kaotik üreteçler ve kanal gürültü seviyeleri altındaki benzetimleri için Matlab ortamında grafik tabanlı bir arabirim tasarlanmıştır [34]. Tasarlanan bu birim sayesinde kullanıcı seçmiş olduğu herhangi bir analog kaotik sistemden farklı sayısal modülasyon teknikleri ve değişken kanal gürültüsü altında oluşturduğu haberleşme sisteminin performansını gözlemleyebilmektedir.

Kaosun kullanıldığı sayısal tabanlı kripto sistemler tezin bundan sonraki kısmında temel ilgi odağı olacaktır. Bu çalışmada tasarlanacak olan sayısal kripto sistem simetrik algoritma içermekte olup, bu sisteme ait anahtar üreteci kaotik harita tabanlıdır.

3.1.2 Kaos Tabanlı Sayısal Kripto Sistemler

Bu kripto sistemler temelde sayısal devreler veya bilgisayar sistemlerinde uygulanan algoritmalardır. Bu algoritmalar sayısal sinyaller üreten kaotik fonksiyonların tekrar eden hesaplamalarına dayanmaktadır. Daha sonra temel kriptografik işlemler (karıştırma ve dağılma) kullanılarak kaotik sinyaller ile bilgi mesajı gizlenir. Bu kripto sistemler, bir veya birden fazla kaotik sistem içerebilir ve algoritmalarda kullanılan kaotik sistemlere ait başlangıç koşulları ve kontrol parametreleri kripto sistemin gizli anahtarları olarak kullanılabilir [41]. Kaos tabanlı analog ve sayısal kripto sistemler ile ilgili özet açıklamalar Tablo 3.2' de verilmiştir.

Tablo 3.2 Kaos tabanlı analog ve sayısal kripto sistemlerin karşılaştırılması

KATEGORİ	YÖNTEM	AÇIKLAMA	
Analog Kripto sistemler	Ekleyici kaos maskeleme	Kaotik sinyal bilgi mesajına eklenir.	
	Kaos kaydırmalı anahtarlama	Sayısal bilgi sinyali farklı kaotik sistemler arasında anahtarlama yaparak bilgi mesajına eklenir.	
	Kaotik modülasyon	Bilgi sinyali kaotik vericinin parametrelerini veya faz uzayını değiştirmek için kullanılır.	
	Kaotik kontrol	Bilgi mesajı klasik yolla şifrelenir ve kaotik sistemi karıştırmak için kullanılır.	
Sayısal Kripto sistemler	Akış şifreleyiciler	Kaotik PRNG	Bilgiye yönelik XOR işlemi için kaotik sözde-rastgele seriler üretir.
		Kaotik ters sistem yaklaşımı	Bilgi mesajı, şifrelenmiş mesaj sinyali ile beslenen kaotik sinyalin çıkışına eklenir.
	Blok şifreleyiciler	Geriye yineleme	Açık bilginin blok kısmı, ters kaotik sistem algoritması kullanılarak şifrelenir.
		İleriye yineleme	Şifreli bilginin blok kısmı kaotik sistemden üretilmiş sözde-rastgele permütasyonlarından elde edilir.
		S- Kutuları	S-kutusu kaotik bir sistemden oluşturulur. Dinamik veya statik bir S-kutusu olabilir.
	Diğer	Kaotik şifreleyici tabanlı araştırmalar	Karakter tablosu kaotik bir sistemden elde edilir. Tablo bilgi mesajının karakterlerini şifrelemek için kullanılır.

Kaos tabanlı kriptografik verimli şifreleme algoritması ilk defa 1989 yılında Matthew tarafından sunulmuştur. Bu tarihten sonra kaotik şifreleme yapıları üzerine olan araştırmalar her geçen gün artmıştır. 1991 yılında, Habustsu ve diğerleri, bu kategoride bilinen ilk ayrık kaotik haritalar tabanlı şifreleme yapısını önermiş ancak bu şifreleme sisteminin kripto analizi Biham tarafından kolayca gerçekleştirilmiştir [44]. Daha sonra, bir veya birden fazla kaotik harita içeren anahtar akışı üreteç tabanlı birçok akış şifreleyiciler ortaya atılmış ve bu alanda çok fazla sayıda araştırma makalesi kaos tabanlı kriptografik algoritmaların geliştirilmesine yönelik sunulmuştur [13]. Ancak, günümüzde mevcut olan kaos tabanlı kriptografik araştırma ve çalışmaların bazıları kriptografik saldırıların temelini dikkate almamıştır. Örneğin, sunulan çalışmada [45] yazarlar, bir boyutlu kaotik bir haritanın kullanıldığı ikili görüntü kodlama algoritmasının [46] yeterli seviyede güvenlik sağlamadığını ve aynı sorunun bir başka çalışmada da [47] olduğunu göstermiştir. Bir başka çalışmada [48] ise tek boyutlu kaotik sistemlerde bulunan küçük anahtar aralığı ve zayıf güvenlik gibi sakıncaların iyileştirilerek üst düzey güvenlik ve kabul edilebilir verimlilik sağlayan yeni bir doğrusal olmayan kaotik bir algoritma sunulmuştur **[**3**].** Bu çalışmalardan anlaşılabileceği üzere kaos tabanlı şifreleme algoritmaları güvenlik açısından iyileştirme ve geliştirmeye açık yapılardır. Dolayısıyla benzer tüm algoritmaların bütünüyle kusurlu olduğu kesinlikle söylenemez.

Kaos tabanlı sayısal şifreleme sistemleri alanında uygulamaya yönelik gerçekleştirilen çalışmalar da mevcuttur. Örneğin, bu çalışmaların birinde [49] cep telefonu uygulamaları için kaos tabanlı kısa mesaj servisi şifreleme yapısı geliştirilmiştir. Burada kullanılan şifreleme yapısı, geliştirilmiş A5/1 algoritması ile birleştirilmiş ve test amaçlı olarak FPGA ortamında uygulanmıştır. Uygulamalar sonunda elde edilen analiz sonuçları, kullanılan kaos tabanlı kısa mesaj servisinin mobil uygulamalar için güvenli olduğunu göstermiştir. Burada sunulan yüksek hızlı kaotik SMS şifreleme yapısı düşük bellek kapasitesi gerektirmekle beraber çok daha yüksek güvenlik sağlamaktadır. Şifreleme hızı ve yöntemin güvenliği yüksek olmakla beraber sunulan bu kripto sistem büyük anahtar alanına sahiptir ve donanımsal gerçekleştirilmesi kolaydır. Bu özellikler aynı zamanda kablosuz ağ uygulamasında kullanılan endüstriyel kontrol gereksinimlerini de karşılamaktadır. Bir başka çalışmada [50], kişiden kişiye göre değişen elektrokardiyogram (EKG) sinyalleri bilgi güvenliğinde biyometrik tanımlamada yeni bir yaklaşım olarak kullanılmıştır. Bu çalışmada şifreleme sistemi, taşınabilir alet kullanarak şifrelemeyi gerçekleştiren kişinin önce EKG sinyallerini toplamakta ve daha sonra kaos teorisine

dayalı akıllı bir algoritma ile şifrelemede kullanılmak üzere Lojistik harita için başlangıç anahtarları üretmektedir. Böylece kullanıcının, kaotik fonksiyon için başlangıç değeri belirlemesine gerek kalmaz. Sunulan kripto sistemine ait benzetim sonuçları sistemin verimliliğini, güvenliğini ve uygulanabilirliğini göstermiştir. Ayrıca şifreleme zamanı kabul edilebilir bir aralıkta olmakla beraber şifrelenmiş bilgi ve kaynak bilgi de aynı büyüklüktedir. Verilen örneklerden de görülebileceği gibi önemli uygulama alanları içerisinde bilgi güvenliğinde kaos kullanılmaktadır. Geleneksel kripto sistemler ile kaos tabanlı kripto sistem yapıları arasındaki fark Tablo 3.3' de gösterilmektedir.

Tablo 3.3 Kaos tabanlı kripto sistemler ile geleneksel kripto sistemlerin karşılaştırılması

Kaos-tabanlı Kripto sistemler	**Geleneksel Kripto sistemler**
Ondalıklı sayı aritmetiği	Tam sayı aritmetiği
Yavaş hesaplama	Hızlı hesaplama
Doğrusal olmayan fonksiyon tabanlı	Genellikle modüler fonksiyon tabanlı
Asal sayılar gerektirmez	Genellikle asal sayılar gerektirir
İstatistiksel sapma mevcuttur	İstatistiksel sapma yoktur

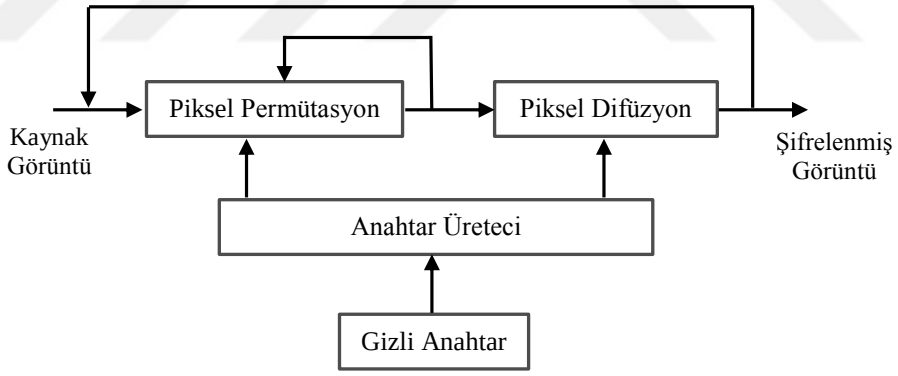
Kaotik dinamiklerin kriptografide kullanılmasına yönelik iki yaklaşım bulunmaktadır. Bunlardan birincisi kaotik sistemleri, sözde-rastgele seriler elde etmek için kullanır ki bunlar bilgi mesajının gizlenmesi için gerekli olan şifreleme anahtarıdır. Diğer yaklaşım ise bilgi mesajı kaotik sistem için başlangıç durumu olarak kullanılır ve şifrelenmiş bilgi oluşturulan yörünge üzerinden takip eder. Buradaki ilk yaklaşım akış şifrelemeye, ikinci yaklaşım ise blok şifrelemeye karşılık gelmektedir. Bu çalışmada geliştirilecek olan görüntü şifreleme algoritması birinci yaklaşımı içermektedir.

3.2. Kaos Tabanlı Görüntü Şifreleme

Görüntü şifreleme mühendislik, endüstriyel ve tıbbi işlemlerdeki uygulamalar için bilgi güvenliğinin önemli bir parçasıdır. Görüntü bilgileri içerisindeki gereksiz fazlalığın, büyük veri kapasitesinin, pikseller arasındaki düşük entropi ve yüksek korelasyon gibi özellikler nedeniyle mevcut şifreleme standartları görüntü şifreleme için uygun değildir. Bunun yanında bu algoritmalar şifreleme süresince beklenenden çok daha fazla hesaplama süresi ve güç gerektirmektedir [51-55]. Kaos tabanlı görüntü şifreleme yapıları, mevcut yöntemler ile karşılaştırıldığında çok daha üstün performans sergilemiştir. Bu anlamda,

kaotik sistemler görüntü şifreleme sistemlerinin güvenliğini geliştirme potansiyeline sahiptirler.

Kaotik harita tabanlı görüntü şifreleme teknikleri temel olarak permütasyon ve difüzyon işlemlerine dayanır. Karıştırma olarak da tanımlanan permütasyon aşamasında kaynak görüntüye ait piksel pozisyonları yer değiştirilir ancak değerlerinde bir değişim yapılmadığından oluşan karışmış görüntü ile kaynak görüntünün histogramları tamamen aynı olur. Dolayısıyla bu yöntem istatistiksel saldırılara karşı tek başına savunmasızdır. Difüzyon aşamasında ise kaynak görüntüye ait piksel değerleri belirli bir algoritmayla değiştirilir. Difüzyon yöntemlerinin çoğunda şifreleme işlemi kaotik serilerin doğrudan görüntü piksel değerleri üzerine yayılarak gerçekleştirilir. Difüzyon, permütasyon işlemine göre yüksek güvenlik sağlamaktadır ancak tek başına kullanıldığında şifreleme gücü yeterli seviyede değildir. Sonuç olarak uygulama çalışmalarında permütasyon ve difüzyon işlemleri yüksek seviye güvenlik sağlaması amacıyla genellikle birleştirilir [56]. 1998 yılında ilk defa görüntü şifreleme için permütasyon ve difüzyon yapısı Fridrich tarafından sunulmuştur. Bu algoritmaya ait şema Şekil 3.4' de verilmiştir.



Şekil 3.4 Fridrich görüntü şifreleme yapısı

Bu yapı içerisinde öncelikle kaynak görüntü, permütasyon aşamasında iki boyutlu alan koruyucu kaotik bir harita ile karıştırılarak görüntü pikselleri arasındaki güçlü bağıntı kırılır. Daha sonra difüzyon aşamasında belirli bir kaotik harita tarafından üretilmiş gizli anahtar kodları ile piksel değerleri sırayla değiştirilir. Bu mimari yapı daha sonradan önerilen hemen hemen tüm kaos tabanlı görüntü kripto sistemlerinin temelini oluşturur [57].

Görüntü şifreleme algoritmalarındaki kaotik sistemler, bir boyutlu ve çok boyutlu kaotik haritalar olmak üzere iki kategoriye ayrılmaktadır. Karmaşık yapısı ve çoklu parametrelere sahip olmasından ötürü çok boyutlu kaotik haritalar görüntü güvenliğinde sıkça kullanılmaktadır. Ancak yapı itibariyle bu sistemlerin donanım ve yazılım ortamındaki uygulamaları çok zor olup, karmaşık hesaplamaya dayalıdır. Diğer bir yandan bir boyutlu kaotik haritalar basit ve kolay uygulanabilir bir yapıya sahiptir. Ancak tek boyutlu bu sistemler aynı zamanda bazı problemler ve eksiklikler içermektedir. Sınırlı veya sürekli olmayan kaotik davranış sergilemeleri, hesaplama analizine karşı dayanıksız olmaları ve sistem çıkışında düzenli olmayan seri dağılımına sahip olmaları bu sistemlerin ciddi kusurlarıdır. Örneğin, Lojistik harita gibi basitliğin ve üst düzey verimliliğin sahip olduğu bir boyutlu kaotik sistemler teorik çalışmalarda ve pratik uygulamalarda yaygın olarak kullanılmıştır. Ancak bu sistemlerin küçük anahtar alanına ve zayıf güvenliğe sahip olması rahatsız edicidir [4]. Bu nedenle, daha iyi kaotik performans sergileyen yeni kaotik sistemlerin geliştirilmesi bu anlamda önemli ve gereklidir [38]. Bu çalışmada, standart Lojistik haritanın sahip olduğu mevcut kusurlar farklı bir teknik ile iyileştirilerek, bir boyutlu yeni bir kaotik harita modeli tasarlanacak ve bu sistem sunulan görüntü kripto sisteminde anahtar üreteci olarak kullanılacaktır.

Son zamanlarda farklı birçok kaos tabanlı şifreleme algoritması görüntü şifrelemedeki problemlere çözüm olması amacıyla bu alanda önerilmiştir. 2010 yılında, Yoon ve Kim [58] başlangıçta küçük bir matrisin lojistik harita kullanarak oluşturduğu bir kaotik görüntü şifreleyici geliştirmişlerdir. Bu çalışmada, oluşturulan bu küçük matrisler ile daha büyük bir permütasyon matrisi geliştirilmiş ve daha sonra bu permütasyon matrisi şifreleme yapısında kaynak görüntüye ait piksellerin yer değiştirilmesinde kullanılmıştır. Bunun yanında, Ismail ve diğerleri **[**59], iki kaotik lojistik haritanın ve 104-bit uzunluğunda harici gizli bir anahtarın kullanıldığı yeni bir kaotik görüntü şifreleyici geliştirmiştir. Her iki lojistik harita için kontrol parametreleri, kullanılan bu harici anahtardan elde edilmiş ve ayrıca sistemi daha güvenli bir yapıya büründürmek için görüntü şifreleyici içerisinde geri besleme mekanizması kullanmışlardır. Bir başka çalışmada, 2011 yılında, Jolfaei ve Mirghadri [10], Baker haritası ile Musa ve diğerlerinin [60], 2003 yılında geliştirdiği AES algoritmasının değiştirilmiş versiyonunu kullanan piksel karıştırmaya dayalı kaotik görüntü şifreleyici ortaya atmışlardır. Nayak ve diğerleri [61], lojistik harita sistemini kullanan ve piksel permütasyonu için bu sistemden üretilen kaotik serilerin dizin konumu esas alan bir kaotik görüntü şifreleyici sunmuşlardır. Bunun

dışında, Xin ve diğerleri [62], karıştırma ve dağılma işlemlerinden oluşan kaos tabanlı bir görüntü kripto sistemi sunmuşlardır. Bu çalışmada, kripto sistemin güvenliğini arttırmaya yönelik olarak karıştırma kısmında, bit seviyesinde permütasyon algoritması kullanan bir yayılma tekniği kullanılmıştır. Bu teknik, kaynak görüntüde 8-bit ile temsil edilen tüm piksel değerlerinin '1' veya '0' olan bit indekslerinin Arnold permütasyon yöntemiyle yer değiştirmesine dayalıdır. Bir başka çalışmada, Peng ve diğerleri [63], kaotik dinamiğe sahip 3-boyutlu Chen sistemi tabanlı ve 192-bit anahtar uzunluğuna sahip bir blok şifreleme algoritması sunmuşlardır. Sathishkumar ve diğerlerinin [64] sunduğu bir diğer çalışmada ise dairesel haritalama tabanlı çoklu kaos dinamiği kullanan yeni bir görüntü şifreleme yapısı anlatılmıştır. Bu çalışmada kullanılan her bir haritanın başlangıç koşullarına bağlı olarak harita yörüngelerinden rastgele sayılar üretilmekte ve bu sayılar arasından şifreleme algoritması için belirli sayıda gizli anahtar seçilmektedir. Daha sonra kaynak görüntü karıştırılıp farklı alt bloklar halinde bölünerek, pozisyon ve değer permütasyonu çoklu kaotik harita tabanlı bir şifreleme gerçekleşmektedir.

Fridrich' in kaotik görüntü şifreleme yapısını sunduğundan beri gerçekleştirilen birçok kaos tabanlı görüntü şifreleme algoritmasının temeli ya sadece kaynak görüntüyü doğrudan şifreleme ya da kaynak görüntüyü karıştırıp hemen ardından karıştırılmış görüntüyü şifreleme tabanlıdır. Bu amaca yönelik olarak bir veya birkaç kaotik sistem bu algoritma yapıları içerisinde kullanılabilir [13]. Güvenlik ve hız arasındaki denge göz önüne alındığında son on yıl içerisinde bir-boyutlu, iki-boyutlu, daha büyük boyutlu kaotik sistemler veya iç içe geçmiş kaotik harita tabanlı birçok farklı görüntü şifreleme teknikleri sunulmuştur. Buna karşın, Fridrich yapısına benzeyen şifreleme algoritmalarından bazılarının kripto analizi başarılı bir şekilde gerçekleştirilmiştir. Örneğin, Chen ve diğerlerinin [65] sunduğu görüntü şifreleme algoritmasının kripto analizi bu çalışmada [66] gerçekleştirilmiştir. Fridrich yapısına benzerlik gösteren bir diğer şifreleme metodu [67] ise bu çalışmada [68] saldırıya uğramıştır [69]. Bu alanda başarılı bir şekilde kırılan kaos tabanlı görüntü kripto sistemlerin çoğu düşük anahtar aralığı ve zayıf güvenlik mekanizmasına sahiptir. Bu zayıflıklar arasında en ciddi olanı ise sistemin anahtar akışının tamamen şifreleme parametrelerine bağlı oluşudur. Dolayısıyla aynı şifreleme anahtarlarının farklı görüntülerin şifrelenmesinde kullanılması, bilinen-bilgi (known-plaintext) ve seçilen-bilgi (chosen-plaintext) saldırı tekniklerinin uygulanabilmesini mümkün kılmaktadır. Bu açıdan değerlendirildiğinde, iyi bir görüntü kripto sisteminde şifreleme anahtarının kaynak görüntüye de bağlı olması gerekir. Bu tez çalışmasında

tasarlanacak olan görüntü kripto sisteminde, anahtar kodları sistem parametreleri dışında kaynak görüntü bilgisine de bağlı olacaktır. Bu durum aynı zamanda sistemin kripto analizini de güçleştirecektir.

Fridrich iyi bir kaos tabanlı görüntü şifreleme algoritmasının iki aşamadan oluşması gerektiğini açıklamıştır [57]. Bunlardan birincisi kaotik bir harita kullanarak görüntü piksellerin sırasının; diğeri ise yine kaotik harita üzerinden her bir pikselin rengini temsil eden değerin değiştirilmesidir. Bu iki aşama sırasıyla karıştırma ve dağılma işlemlerini göstermektedir. Bu alanda sunulmuş birçok kaos tabanlı görüntü şifreleme yapısı bu iki tekniği içermektedir. Örneğin, Lian ve diğerleri [70] çalışmasında daha büyük anahtar aralığı elde etmek için karıştırma aşamasında geliştirilmiş standart harita sistemini; dağılma işlemi için ise lojistik harita modelini kullanmıştır. Bu iki kaotik haritanın parametreleri her bir iterasyonda oluşturulan bir anahtar akışı ile belirlenmektedir. Bu algoritmayı, daha sonra Wong ve arkadaşları [71], permütasyon kısmında belirli bir dağılma etkisi gösteren, 'ekle-ve-çıkar' operasyonunun sunulduğu bir algoritma yapısını ortaya atarak iyileştirmiştir. Böylelikle şifreleme sırasında zaman harcayan iş yükü de azaltılmıştır. Wang ve diğerlerinin [72] sunduğu bir diğer kaos tabanlı görüntü şifreleme algoritmasında değişken kontrol parametreleri kullanılarak bilinen/seçilen saldırılara karşı direnç gösteren bir şifreleme yapısı ortaya atılmıştır. Fu ve diğerleri [13], çalışmalarında çift yönlü dağılma olarak adlandırılan geliştirilmiş dağılma stratejisini sunmuşlar ve dağılma işleminde yeni bir karıştırma mekanizması ortaya atmışlardır. Böylelikle şifrelemedeki yayılma işlemi önemli ölçüde hızlandırılmış ve daha az şifreleme iterasyonu ile aynı seviye güvenlik elde edilmiştir. Bu çalışmalar dışında, daha büyük anahtar aralığı elde etmek ve tek boyutlu kaotik sistemlerdeki zayıf güvenliği ortadan kaldırmak amacıyla daha yüksek boyutlu kaotik sistemler görüntü şifrelemede kullanılmıştır [73]. Yüksek boyutlu kaotik sistemler dışında çoklu kaotik haritalar ve birleşik kaotik sistemlerden de bu alanda faydalanılmıştır [74]. Bazı araştırmacılar, kaynak görüntünün piksel değerlerini bit seviyesine indirerek bir permütasyon gerçekleştirmişlerdir [75,76]. Bu çalışma aynı zamanda piksel değerini de değiştirdiğinden şifrelemede etkili bir karıştırmayı sunmaktadır.

Bu tez çalışmasında, kullanılacak olan şifreleme algoritması iki kısımdan oluşacaktır. Görüntü piksellerinin pozisyonlarına ait yer değişimi için Arnold kedi haritası (Arnold Cat map) sisteminden faydalanılacaktır. Esas şifrelemenin gerçekleştiği dağılma kısmında ise piksel değerlerinin değişimi için özgün bir difüzyon fonksiyonu kullanılacaktır.

3.2.1 Arnold Kedi Haritası

Rus matematikçi Vladimir Arnold tarafından 1960 yılında, bir kedi görüntüsü üzerinde çalışılırken keşfedilen bu harita sistemi kaotik haritalamanın özel bir durumudur [77]. Bu sistem, bazı kaos ilkelerinin basit gösteriminin güzel bir örneği olarak nitelendirilir ve sahip olduğu etkili karıştırma özelliği ile bu alanda sıkça kullanılmaktadır. Arnold kedi haritası (AKH), 2-boyutlu ayrık zamanlı bir sistem olup matematiksel olarak Denklem (3.2)' deki gibi tanımlanır.

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = C \begin{bmatrix} x \\ y \end{bmatrix} \bmod 1 \tag{3.2}$$

Burada x ve x_{n+1} sırasıyla sistemin mevcut ve bir sonraki durumunu gösterirken, C ise dönüşüm matrisi olup, Denklem (3.3)' de verildiği gibi ifade edilir.

$$C = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \tag{3.3}$$

Görüntü şifrelemede ve açık anahtarlı kripto sistemler gibi pratik uygulamalarda, AKH sistemindeki dönüşüm matrisi genellikle Denklem (3.4)' de verilen ifade ile değiştirilir. Böylelikle sistemden yüksek oranda rastgelelik sağlamak ve bunun sonucunda da daha güvenli bir kripto sistem elde etmek amaçlanır.

$$C = \begin{bmatrix} 1 & p \\ q & pq+1 \end{bmatrix} \tag{3.4}$$

Burada p ve q pozitif tamsayılardır. Ayrıca $|C| = 1$ olması, AKH sisteminin faz uzayında yörünge alanını koruduğunu göstermektedir. Bu özellik ise IR^2 düzleminde herhangi bir görüntünün bu matris ile dönüştürülmesiyle görüntüye ait ilk alanın korunabilir olması şeklinde yorumlanır. Sonuç olarak, AKH sisteminin genelleştirilmiş durumu Denklem (3.5)' deki gibi olur.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & p \\ q & pq+1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \bmod N \tag{3.5}$$

Burada (x, y), $N \times N$ ölçüsüne sahip karasel görüntünün piksel koordinatını; (x', y') ise AKH sonucu oluşan karışmış görüntüye ait yeni koordinatı göstermektedir. Matris formatındaki bu ifade matematiksel olarak Denklem (3.6)' daki gibi de tanımlanabilir.

$$\begin{aligned} x' &= [x + py] \bmod N \\ y' &= [qx + (pq+1)y] \bmod N \end{aligned} \tag{3.6}$$

Denklem (3.6)' da gösterilen ifadede *mod* işlemi sonucunda hesaplanan x' ve y' değerleri 0 ile N-1 arasında sınırlandırılır. Ancak görüntüye ait piksel koordinatı pozisyon olarak 0 olamayacağından dolayı tüm yeni koordinatlara 1 ilave edilerek bu olası durumdan kurtulmak mümkündür. Bu durum, aynı zamanda yer değiştirme sonucunda oluşan piksel pozisyonunu N yapabilecektir. Sonuç olarak AKH sisteminin kriptografik uygulamalar için genelleştirilmiş ifadesi Denklem (3.7)' de verildiği gibi düşünülebilir.

$$\begin{aligned} x' &= [x + py] \bmod N + 1 \\ y' &= [qx + (pq+1)y] \bmod N + 1 \end{aligned} \tag{3.7}$$

Denklem (3.7)' de tanımlanan AKH sistemi, çalışmalarımızda görüntünün piksel koordinatlarının rastgele bir şekilde yeniden sıralanmasını sağlayacak bir yöntem olarak kullanılacaktır. Bu yöntemin bir diğer önemli özelliği ise *Poincare* yinelenme teoremini desteklemesidir. Bu teori kısaca belirli sistemlerin yeterli bir süre sonunda tekrar başlangıç durumuna dönmesini açıklamaktadır. Sonuç itibariyle AKH kullanılarak belli bir iterasyon sayısı sonunda, orijinal görüntüdeki ilk piksel pozisyonlarına tekrar ulaşılır. Bu aynı zamanda sistemin periyodik özelliğe sahip olduğunu da göstermektedir. Burada bir iterasyon, görüntüye ait piksellerin tamamının bir defa yer değiştirmesi olarak tanımlanır.

AKH sistemindeki p ve q parametreleri birer şifreleme anahtarı gibi düşünüldüğünde bu durum sistemin anahtar aralığını arttırmaya yardımcı olarak dış saldırılara karşı dayanıklılığı arttıracak ve aynı zamanda daha uzun bir periyot süresi sağlayarak sistem rastgeleliğini geliştirecektir. Böylelikle, kripto sistemin istatistiksel saldırılara karşı daha

dirençli bir yapıya dönüşmesi beklenir. AKH kullanarak belli bir iterasyon sonunda oluşan karışmış bir görüntü için bu sistemin tersi üzerinde aynı parametreler kullanılarak tekrar orijinal görüntü elde edilebilir. AKH sistemin tersi, Denklem (3.8)' deki gibi tanımlanır.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} pq+1 & -p \\ -q & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix} \bmod N \tag{3.8}$$

3.2.2 Standart Lojistik Harita Sistemi

Standart Lojistik harita (SLH), kaotik davranışın gözlemlenebildiği en basit ayrık zamanlı sistemlerden biridir. Bir boyutlu olan bu harita, doğrusal olmayan dinamik sistemlerde incelenen kaos teorisinin temelini oluşturur ve bu alanda sıkça kullanılan genel bir örnektir. Bu sistem, Denklem (3.9)' daki gibi tekrarlı bir denklem ile tanımlanır [78].

$$x_{n+1} = r.x_n(1-x_n) \tag{3.9}$$

Burada *r* sistemin kontrol parametresidir ve tanım aralığı $r \in (0,4]$ şeklindedir. Denklemde verilen x_n sistemin durum değişkenini gösterirken; x_{n+1} ise bir sonraki çıkış değerini ifade eder. Sistemin durum değişkeni, $x_n \in (0,1)$ olarak tanımlıdır ve reel bir değere sahiptir. Sistemin mevcut çıktısı x_{n+1}, bir önceki x_n değerine bağlı olmakla beraber sistemden elde edilen tüm seriler de sistemin başlangıç değeri olan x_0' a bağlıdır. Bu ifade Denklem (3.10)' da gösterildiği gibi özetlenebilir.

$$x_n = f(x_{n-1}) = f^n(x_0) \tag{3.10}$$

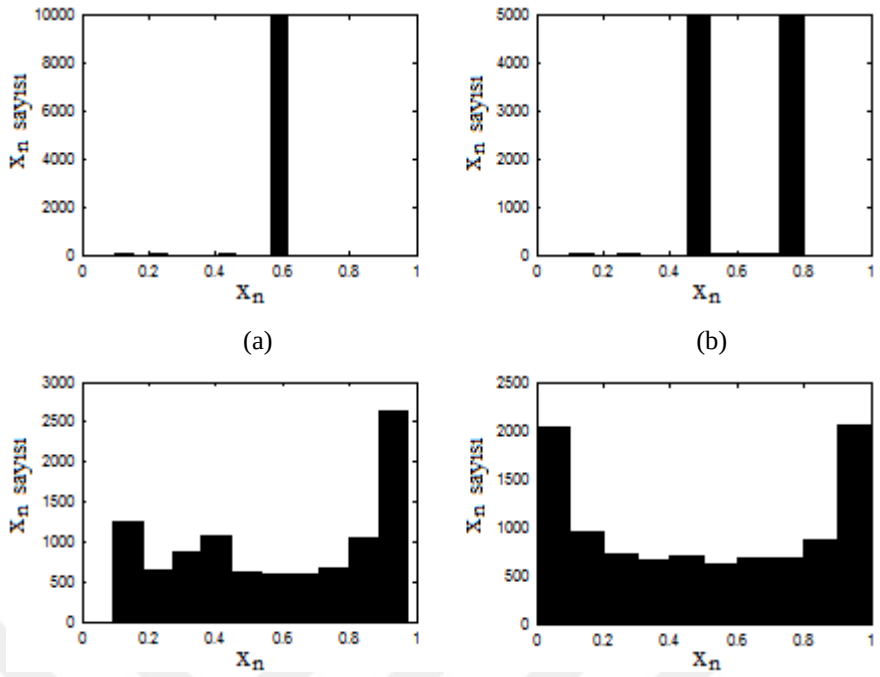
Burada x_n, x_0' ın *n.* iterasyonu olup, tüm iterasyonların kümesi ise fonksiyonun haritası olarak adlandırılmaktadır. Dolayısıyla belirli bir başlangıç değeri ve kontrol parametresi ile bir $\{x_n\}_{n=1}^{\infty}$ serisi hesaplanabilir ve seride oluşan bu tekrarlı çözümler, sistemin yörüngesi olarak ifade edilmektedir.

Bir sistemin aynı başlangıç değeri ile tamamen benzer bir davranış sergileyerek aynı son durumuna ulaşması bu sistemin deterministik yapıya sahip olmasıyla açıklanır. Ancak başlangıç değerinde çok küçük bir değişimin sistemde çok farklı bir son durum yaratması,

sistemdeki kaos varlığı ile açıklanmakta ve deterministik kaos kavramını oluşturmaktadır. Bu anlamda SLH, kaos içeren deterministik bir sistem olarak da yorumlanabilir.

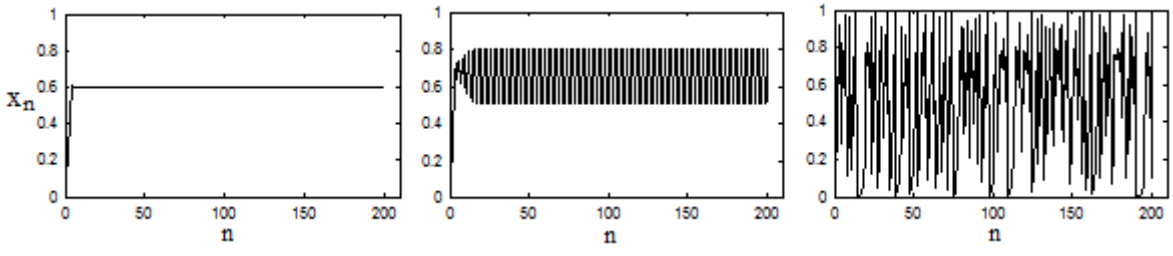
SLH sistemi, esasen sınırlı kaynaklara sahip olan bir büyüme artış modelini temsil etmesine [79] karşın bu sistem, kaos teorisi içerisinde yeni fikirlerin ortaya çıkmasında özellikle kaosun kriptografi uygulamalarına yönelik çalışmalarında yararlı bir model olmaya devam etmektedir [80]. Bunda sistemin hem yazılım hem de donanım ortamında tasarımının kolay oluşu ve sadece basit aritmetik işlemlerle gerçekleştirilmesinin rolü büyüktür. Bunun yanında sistemin kaos durumunda rastgele benzeri bir davranış sergilemesi ve uzun dönemde sistem durumunun öngörülemez oluşu, bu haritanın kripto sistemlerde sözde-rastgele anahtar üreteci olarak kullanılmasına olanak sağlamaktadır. SLH özellikle kaos tabanlı görüntü şifrelemede sıkça kullanılmıştır [24, 72, 76, 81-85].

SLH basit bir matematiksel denkleme sahip olmasına rağmen kontrol parametresine bağlı olarak farklı davranışlar sergileyebilmektedir. Bu dinamik davranışlar değişken r parametresi için, $0 \le r \le 3$ iken sistemin son durumu durağan ve $3 < r < 3.57$ durumunda ise periyodik salınım şeklindedir. Kontrol parametresi $3.57 \le r \le 4$ durumunda ise sistem davranışı karmaşık bir kaotik yapıya dönüşürken x_n serileri düzensiz bir hal alır ve sistem başlangıç değerine karşı aşırı duyarlı olur. Bu durumda seriler, sınırlı bir aralıkta olmasına rağmen periyodik olmayan sözde-rastgele özelliktedir ve herhangi bir iterasyondan sonra yakınsamazlar. SLH sisteminden farklı r değerleriyle elde edilen 10,000 serinin histogram dağılımı Şekil 3.5' de verilmiştir.



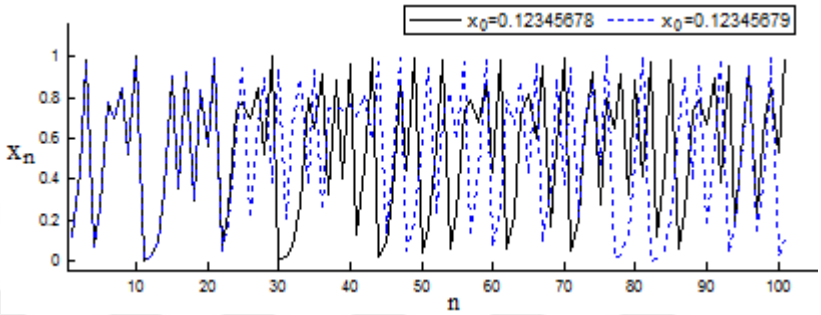
Şekil 3.5 SLH sisteminde farklı *r* değerleriyle üretilmiş serilerin dağılımları; (a) $r = 2.5$, (b) $r = 3.2$, (c) $r = 3.9$, (d) $r = 4$

Kontrol parametresi $r = 2.5$ iken sistemin sabit davranış sergilediği Şekil 3.5(a)' da; $r = 3.2$ durumunda sistemin belli değerler arasında salınım yaparak periyodik bir davranış oluşturduğu Şekil 3.5(b)' de gösterilmektedir. Şekil 3.5(c)' de ise sistem kaos durumunda olmasına rağmen x_n değerleri (0,1) aralığında tam olarak düzenli bir şekilde dağılmamıştır. Grafik sonuçlarından açıkça görüldüğü gibi kontrol parametresi, $r = 4$ durumunda sistem mükemmel kaotik davranış sergilemekte ve çıkış değerleri (0,1) aralığının tamamında simetrik biçimde dağılmaktadır. Şekil 3.6' da ise sistemin farklı kontrol parametrelerindeki yörüngesi verilmiştir. SLH sisteminde sergilenen durağan, periyodik ve karmaşık davranış biçimleri sırasıyla Şekil 3.6(a), Şekil 3.6(b) ve Şekil 3.6(c)' de gösterilmiştir.



Şekil 3.6 SLH sisteminden elde edilen seriler; (a) $r = 2.5$, (b) $r = 3.2$, (c) $r = 4$

Kaotik bir sistemin en belirgin tanımlayıcı özelliği başlangıç koşullarına olan hassas duyarlılığıdır. SLH' de değerce birbirine çok yakın iki başlangıç değeri $x_0 = 0.12345678$ ve $x_0` = 0.12345679$ seçilerek iki kaotik seri elde ediliyor. Daha sonra bu serilerin oluşturduğu yörüngelerin tamamen farklı olduğu ve zamanla birbirinden uzaklaştığı durum Şekil 3.7' de grafiksel olarak gösterilmiştir.



Şekil 3.7 SLH sisteminin başlangıç değerine olan duyarlılığı

Doğrusal olmayan dinamik bir sistemde, birbirine çok yakın olarak başlayan iki yörünge artan zaman ile birlikte birbirinden çok fazla sapıyorsa bu sistem kaotik olarak adlandırılır. Bu yakın yörüngelerin zamanla birbirlerinden ayrıldığı oran, Lyapunov üsteli adı verilen bir değer ile karakterize edilmektedir. Lojistik harita sisteminde, $x_{n+1} = f(x_n; r)$ için Lyapunov üsteli λ, birbirine çok yakın iki farklı başlangıç değeri ile x_0 ve $x_0 + \varepsilon_0$ gerçekleştiğinde, n iterasyon sonunda bu iki nokta arasındaki uzaklaşma yaklaşık olarak $|\varepsilon_n| \approx |\varepsilon_0| e^{n\lambda}$ denklemi ile ifade edilir. Daha açık bir şekilde, her iki tarafın doğal logaritması alındığında,

$$\lambda \approx \frac{1}{n} \ln \left| \frac{\varepsilon_n}{\varepsilon_0} \right| = \frac{1}{n} \ln \left| \frac{f^n(x_0 + \varepsilon_0) - f^n(x_0)}{\varepsilon_0} \right| \tag{3.11}$$

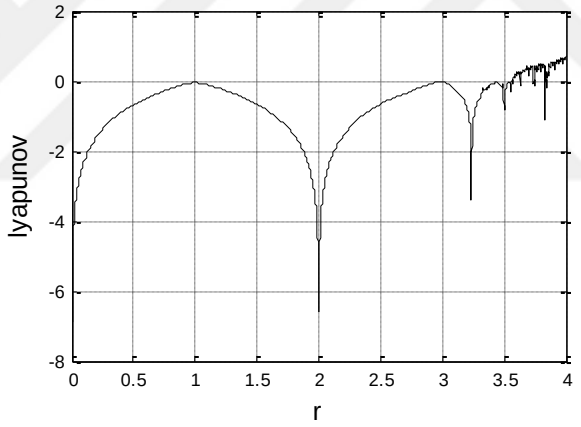
Denklem (3.11) elde edilir ve çok küçük ε_0 değerleri için logaritma içerisindeki kalan terim düzenlenirse,

$$\lambda \approx \frac{1}{n} \ln \left| \prod_{i=0}^{n-1} f'(x_i) \right| = \frac{1}{n} \sum_{i=0}^{n-1} \ln |f'(x_i)| \tag{3.12}$$

Denklem (3.12)' de verilen ifade bulunur. Sonuç olarak $n \to \infty$ durumu için, ayrık zamanlı sistemlerde Lyapunov üsteli Denklem (3.13)' deki gibi hesaplanır.

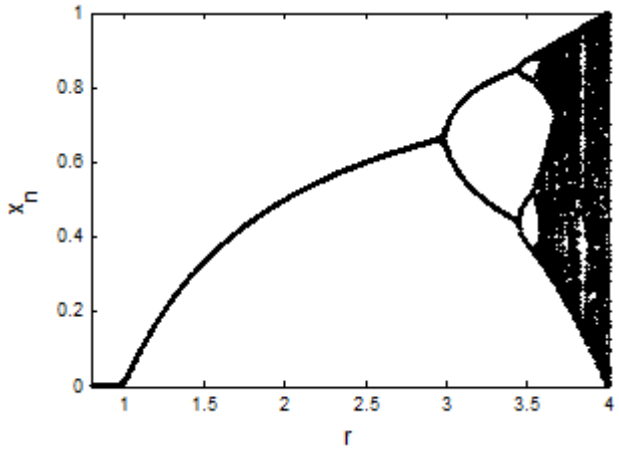
$$\lambda = \lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} \ln\left| f'(x_i) \right| \tag{3.13}$$

Burada, $\lambda > 0$ ise komşu yörüngeler birbirinden üstel olarak ayrılır ki bu durum kaos dinamiğine karşılık gelmektedir. Diğer bir yandan yörüngeler sabit bir noktaya veya limit döngüsüne doğru yakınsıyor ise $\lambda < 0$ olur. Dolayısıyla, bir sistemin kaotik durumda olup olmadığı, Lyapunov üstelinin işareti ile belirlenebilir. Sistemin λ değerinin, r parametresine göre grafiksel gösterimi Lyapunov spektrumunda verilir. SLH sisteminin Lyapunov spektrumu Şekil 3.8' de verilmiştir.



Şekil 3.8 SLH sisteminin Lyapunov spektrumu

Lyapunov grafiğinden görülebildiği gibi kontrol parametresinin $r \in [3.57, 4]$ aralığındaki çoğu değerinde sistem kaotik durumdadır ve λ değeri pozitiftir. Sistemin sabit veya periyodik davranış sergilediği durumda ise λ değeri negatiftir. Dinamik bir sistemin sabit bir noktadan kaotik durumuna kadar olan tüm davranışlarının, değişken kontrol parametresine göre gözlemlenmesi çatallaşma (Bifurcation) diyagramı olarak adlandırılan grafikte verilir. Çatallaşma diyagramı sistem davranışının analizi için kullanılır ve sistemin dinamik durumu hakkında bilgi verir. SLH sistemine ait çatallaşma diyagramı Şekil 3.9' da gösterilmiştir.



Şekil 3.9 SLH sisteminin çatallaşma diyagramı

Çatallaşma diyagramından görüldüğü gibi kontrol parametresi 0 değerine doğru yaklaştıkça sistem durağan bir yapıya; 4' e doğru yakınlaştıkça karmaşık bir yapıya dönüşmektedir. Ayrıca grafikte $r \in [3.57, 4]$ için sistemdeki x_n serileri bu bölgede düzensiz bir şekilde dağılmaktadır. Bu tanım aralığındaki çoğu değerde SLH sistemi kaotik davranış sergilemektedir ancak belli bazı noktalarda, örneğin $r \approx 3.83$ gibi sistemin Lyapunov üsteli Şekil 3.8' de görüldüğü gibi negatif olmakta ve bu noktada sistem kaos özelliği taşımamaktadır. Sistemin çatallaşma diyagramından da kolayca görülebilen bu kısım literatürde açık bölge (blank zone) olarak adlandırılmaktadır [86]. Sonuç olarak kontrol parametresi, $r \in [3.57, 4]$ aralığında olsa bile tüm r değerleri altında sistem sürekli kaotik davranış sergilememektedir. Sistem davranışındaki bu kaos eksikliği SLH sisteminin ciddi bir kusurudur. Bu açıdan sistemin iyi bir kaotik özellik göstermesi kontrol parametresinin $[3.9, 4]$ alanına düşürülmesiyle mümkün olabilir ancak bu çözüm tek alternatif yöntem de değildir.

Lojistik haritanın bir diğer problemi, kontrol parametresinin sınırlı bir aralıkta olması ve sadece $r = 4$ değeri için mükemmel kaos sergilemesidir. Bir kripto sisteminde şifreleme anahtarlarının rastgele ve homojen bir dağılıma sahip olması istenen özelliklerdir. Böyle bir kripto sistem için anahtar üreteci olarak SLH kullanılacaksa, sistemin sahip olduğu bu kusurlar dikkate alınıp, öncelikle iyileştirilmelidir. Aksi takdirde bu durum, şifreleme anahtarlarından beklenen karakteristik özelliklerin oluşmasını engeller. Bu olumsuzlukların belirmesi ise kripto sistemde küçük anahtar aralığına, zayıf güvenliğe ve verimsiz bir şifrelemeye neden olur. Sonuç olarak, sistemdeki bu ciddi sıkıntıların giderilmesi tüm kripto sisteminin güvenliği açısından çok önemlidir.

SLH sisteminin geliştirilmesine ve iyileştirilmesine yönelik birçok çalışma bu alanda sunulmuştur. Bunların çoğu, SLH denklemine yeni bir ek parametre eklenmesi veya iç içe geçmiş birden fazla kaotik harita kullanılması odaklıdır. Bu çalışmalara ek olarak farklı yöntemler de benimsenmesine karşın hedeflenen amaç aynıdır. Sistemin kontrol parametre aralığının arttırılması ve tüm değerler altında sistemin sürekli kaos durumunda olması, geliştirilen veya iyileştirilen haritadan beklenen özelliklerdir. Böylelikle sistemin kriptografik uygulamalardaki karmaşıklığı dışında yüksek güvenliği de sağlanmış olacaktır. Bundan sonraki kısımda SLH sistemi iyileştirilerek bir boyutlu yeni bir kaotik harita tasarlanacaktır. Bu yeni haritada, SLH sisteminin sahip olduğu tüm kusurlar giderilecek ve sistemde sürekli kaos durumu olması sağlanacaktır.

4. İYİLEŞTİRİLMİŞ LOJİSTİK HARİTA SİSTEMİ

Bu kısımda, SLH denklemine ek bir parametre ilave edilerek düzenlenmiş denklem sistemi ve bu sistemin iyileştirilmesiyle tasarlanan yeni bir kaotik harita modeli detaylı şekilde anlatılacaktır. Bir önceki bölümde bahsedilen SLH sistemindeki eksikliklerin nasıl bir teknik kullanılarak giderildiği ve mevcut sistemin iyileştirilmesi bu bölümün önemli başlıkları olacaktır. Ayrıca yeni sistemin bir önceki sisteme göre sağladığı avantajların önemine vurgu yapılarak, tasarlanan haritanın sunulan kripto sistemde sözde-rastgele kod üreteci olarak kullanılması sağlanacaktır. İstatistiksel test yöntemleri kullanılarak iyileştirilen sistemin karakteristik özellikleri Matlab ortamında detaylı bir şekilde analiz edilip, sistemin kriptografik anlamda kod üreteci açısından uygunluğu incelenecektir.

SLH sisteminin iyileştirilmesine yönelik birden fazla yöntem bu alandaki bilimsel çalışmalarda sunulmuştur. Bu kısımda, mevcut SLH denklemine yeni bir parametre ilave edilerek sistemin performansının iyileştirilmesi ve kontrol parametresinin tanım aralığının arttırılması hedeflenmiştir. Bu amaca yönelik olarak öncelikle SLH sistemine ek parametre ilave edilerek oluşturulan matematiksel sistem, Denklem (4.1)' deki gibi gibi ifade edilsin.

$$x_{n+1} = \lambda . x_n (1 - x_n) + p \tag{4.1}$$

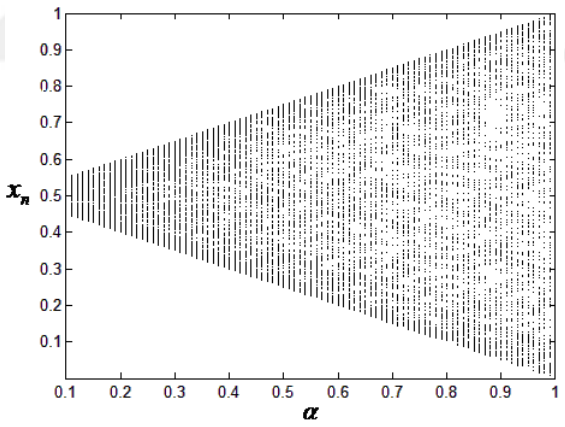
Burada durum değişkeni x_n, $\left[\frac{1-\alpha}{2}, \frac{1+\alpha}{2}\right]$ aralığında sınırlı ve $0 < \alpha < 1$ olacak şekilde tanımlansın. Denklem (4.1)' deki p, sisteme ilave edilen ek bir parametre olup, değerce α parametresine bağlıdır. Bu durumda, yeni denklemden elde edilebilecek maksimum değer $x_n = 0.5$ noktasında oluşur ve değeri $\frac{\lambda}{4} + p$ olur. Minimum ise $x_n = \frac{1-\alpha}{2}$ noktasında $\lambda.(\frac{1-\alpha}{2}).(\frac{1+\alpha}{2}) + p$ değeri ile oluşur. Bu durum, Denklem (4.2)' deki gibi özetlenebilir.

$$\begin{aligned} \frac{1-\alpha}{2} &= \lambda.(\frac{1-\alpha}{2}).(\frac{1+\alpha}{2}) + p \\ \frac{1+\alpha}{2} &= \frac{\lambda}{4} + p \end{aligned} \tag{4.2}$$

Denklem (4.2)' de verilen eşitlikler çözüldüğünde λ , p parametreleri sırasıyla $\lambda = \frac{4}{\alpha}$ ve $p = \frac{\alpha^2 + \alpha - 2}{2\alpha}$ olarak bulunur. Bu değerler, Denklem (4.1)' de yerine yazılırsa Denklem (4.3)' de verilen ifade elde edilir.

$$x_{n+1} = \frac{4.x_n(1-x_n)-1}{\alpha} + \frac{1+\alpha}{2} \tag{4.3}$$

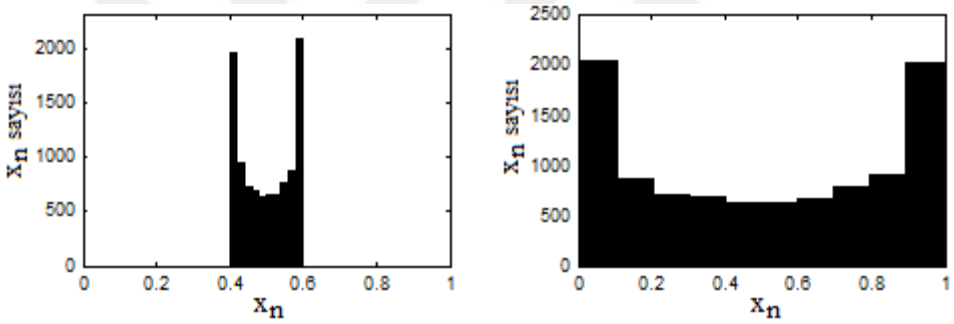
Denklem (4.3)' de verilen sistem, SLH sistemine göre biraz daha karmaşık bir yapıya sahiptir. Burada α parametresi, tanım aralığı gereği 1 ile sınırlandırılmıştı ve dikkat edilirse $\alpha = 1$ değerinde Denklem (4.3), SLH sistemine geri dönmektedir. Burada α parametresi ile x_n serilerinin dağılımı arasındaki ilişkiyi görebilmek için farklı α değerleri altında sistemden üretilen x_n serilerinin dağılım aralığı hesaplanmış ve oluşturulan çatallaşma grafiği Şekil 4.1' de verilmiştir.



Şekil 4.1 x_n aralığının α değerine göre değişimi

Bir kripto sistemde üretilen gizli anahtar kodları, periyodik olmamalı ve her bir anahtar elemanı dağılım içerisinde olabildiğince eşit olasılıkta oluşmalıdır. Sunulan kripto sistemde x_n serilerinin şifreleme kodu olarak kullanılmasından dolayı tanım aralığı içerisinde olabilecek tüm değerlere sahip olması ve bu aralıkta homojen bir şekilde dağılması önem arz edeceğinden yeni sistemdeki x_n dağılımının sürekli olarak büyük aralıkta tutulması gerekir. Şekil 4.1' deki grafikten anlaşılabileceği gibi α değeri,

maksimum 1 değerine doğru yaklaştıkça sistemdeki x_n aralığının arttığı; sıfıra doğru yaklaştığında ise bu aralığın azaldığı görülmektedir. Örneğin, α parametresi 0.2 gibi bir değer seçildiğinde, sistemden elde edilen x_n serileri 0.4 ile 0.6 arasında dağılacaktır. Bu karakteristik özellik, iyi bir anahtar üretecinden beklenmeyen bir durumdur. Ayrıca sistemden elde edilen x_n serilerinin, tüm α değerlerinde 0 ile 1 arasında düzenli bir şekilde dağılması yani homojenlik sergilemesi çok önemlidir. Şekil 4.2(a) ve Şekil 4.2(b) sırasıyla $\alpha = 0.2$ ve $\alpha = 0.99$ için x_n serilerinin dağılım aralığını göstermektedir. Şekil 4.2(a)' da görüldüğü gibi x_n serileri sınırlı bir aralıkta olup, sistemden belli değerler dışında çıkış alınamamaktadır. Sonuç olarak, tasarlanacak sistem için tüm kontrol parametresi altında üretilecek serilerin dağılımı Şekil 4.2(b)' deki gibi olması istenilen bir sonuç olacaktır.



(a) (b)

Şekil 4.2 Denklem (4.3)' deki x_n aralığının değişimi; (a) $\alpha = 0.2$, (b) $\alpha = 0.99$

Kripto sistemde α, şifreleme algoritmasına ait gizli bir anahtar parametresi olarak düşünülmüştür. Bu değerin olabildiğince geniş bir spektrumda yayılması, kripto sistemdeki anahtar aralığının artmasına yardımcı olacaktır. Daha önceden de belirtildiği gibi Denklemi (4.3)' de verilen sistemde, kontrol parametresinin tanım aralığı $\alpha \in (0,1]$ şeklindeydi. Burada α parametresinin sahip olduğu tanım aralığının arttırılmasına yönelik olarak oluşturulmuş denklem sistemine yeni bir s parametresi daha ilave ederek, bu parametrenin sınırlı olmayan bir spektrumda yayılması hedeflenmiştir. Böylelikle sistemin etkili anahtar aralığı önemli ölçüde arttırılmış olacaktır. Bu anlamda α parametresi ile ilgili olarak Denklem (4.4)' de verilen dönüşüm, Denklem (4.3) için uygulanmıştır.

$$\alpha \Rightarrow \frac{\alpha}{s} \tag{4.4}$$

Burada s yeni sistemin öncelikli kontrol parametresi olup, değer olarak 1' den farklı pozitif bir sayıyı göstermektedir. İlgili dönüşüm yapıldıktan sonra iyileştirilen yeni haritaya ait denklem sistemi Denklem (4.5)' deki gibi oluşur.

$$x_{n+1} = \frac{4.s.x_n(1-x_n)-s}{\alpha} + \frac{s+\alpha}{2s} \tag{4.5}$$

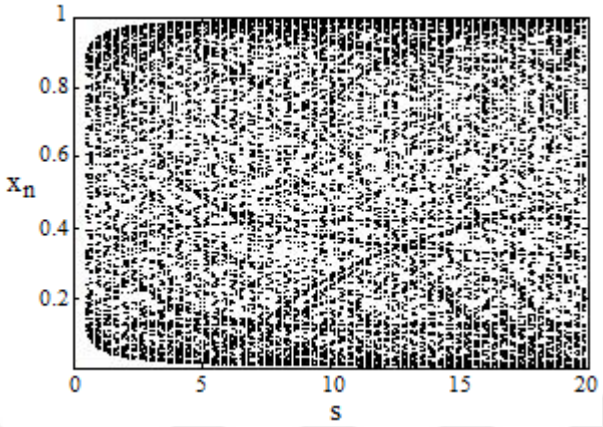
Denklem (4.3)' de verilen sistemde, α parametresinin tanım aralığı $0<\alpha<1$ şeklinde ve x_n ise $\left[\frac{1-\alpha}{2},\frac{1+\alpha}{2}\right]$ aralığında sınırlandırılmıştı. Dönüşüm sonrası yeni sistemdeki α parametresinin tanım aralığı $0<\alpha<s$ şeklinde; x_n serisinin yeni aralığı ise $\left[\frac{s-\alpha}{2s},\frac{s+\alpha}{2s}\right]$ gibi olacaktır. Denklem (4.5), tek boyutlu yeni bir kaotik haritanın matematiksel modeli olup, SLH sistemine göre fazladan bir parametreye daha sahiptir. Çalışmamızın bundan sonraki kısımlarında bu sistem, iyileştirilmiş Lojistik harita (İLH) olarak adlandırılacaktır.

İLH sisteminde s parametresi, α parametresinden değerce büyük olması gerekir aksi takdirde üretilen x_n serisi (0,1) tanım aralığının dışında oluşacaktır. Ayrıca $\alpha = s$ durumunda, sistemin SLH denklemine geri döndüğü açık bir şekilde görülmektedir. İLH sisteminde durum değişkenin, $x_n \in \left[\frac{s-\alpha}{2s},\frac{s+\alpha}{2s}\right]$ olmasından dolayı α parametresinin s parametresine göre yakınlığı, sistem çıkışı x_n' in oluşma aralığını belirleyecektir. Bu ifade matematiksel olarak incelendiğinde,

$$\begin{aligned} &\lim_{\alpha \to s} \frac{s-\alpha}{2s} = 0 \\ &\lim_{\alpha \to s} \frac{s+\alpha}{2s} = 1 \end{aligned} \tag{4.6}$$

Denklem (4.6)' da verilen limit ifadelerinden α parametresi, s değerine yaklaştıkça x_n aralığının artacağı; uzaklaştığında ise bu aralığın azalacağı görülebilmektedir. SLH sisteminde olduğu gibi İLH sisteminde de $x_n \in (0,1)$ olmalıdır. Yeni sistemden farklı s

değerleri altında elde edilen x_n serilerinin dağılma aralığını gösteren çatallaşma diyagramı Şekil 4.3' de verilmiştir.

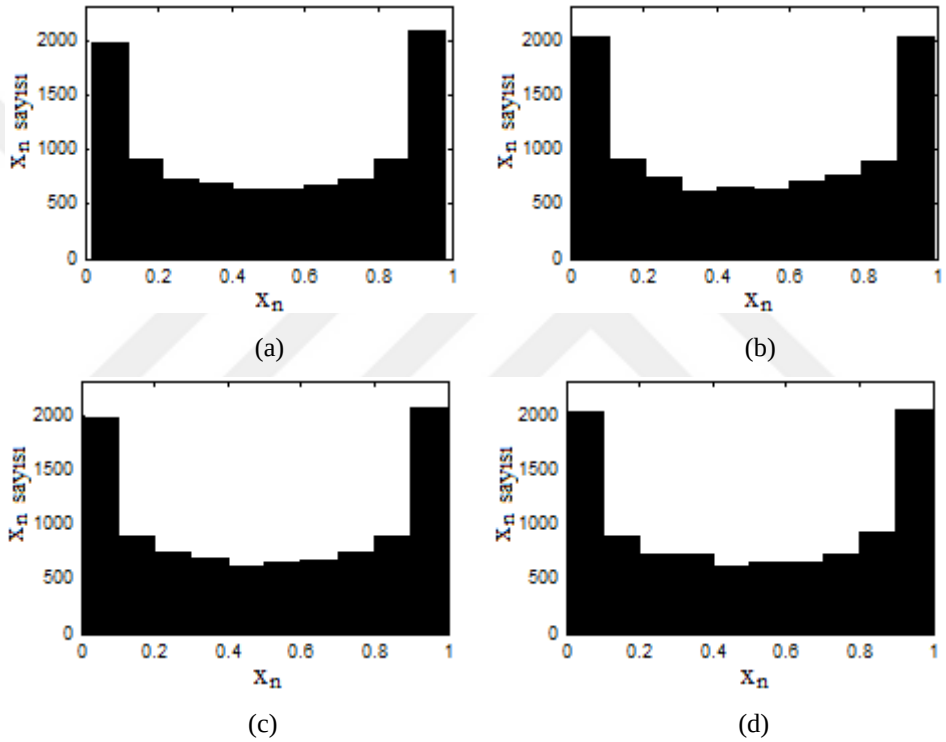


Şekil 4.3 İLH sisteminin çatallaşma diyagramı

Şekil 4.1' de çatallaşma diyagramı verilen kısmen iyileştirilmiş sistemde x_n serilerinin oluşma aralığı, α değeri arttıkça doğrusal bir şekilde artıyor; azaldıkça ise bu aralık azalıyordu. Sistemdeki bu özellik, homojen ve düzenli bir dağılımın oluşmasını engeller niteliktedir. Şekil 4.3' de ise verilen grafiksel sonuçtan gözlemlenebildiği gibi İLH sisteminde hemen hemen tüm s değerleri altında x_n serileri, beklenildiği gibi (0,1) aralığının tamamında yayılarak tüm alanı kaplamaktadır. Harita denklemine eklenen s parametresi herhangi bir üst değer ile sınırlandırılmadığı gibi tanım aralığı da $s \in (1,\infty)$ olacaktır. Ayrıca, bütün s parametre değerleri için sistem benzer bir davranış sergilemektedir. Sergilenen bu davranışın sürekli kaotik olması sistemden beklenilen en önemli karakteristik davranıştır. Daha önceden de belirtildiği gibi İLH sisteminde α değeri, s değerine ne kadar yakınsa üretilen x_n serilerinin aralığı o kadar büyük olacaktır. Buna yönelik olarak üretilen x_n serilerinin (0,1) aralığının tamamında yayılması amaçlandığından bu çalışma içerisinde α değeri, sistemin başlangıç değeri x_0 kadar s değerine yakın tutulması tercih edilmiştir. Dolayısıyla s ile α arasında Denklem (4.7)' deki gibi bir ilişki belirlenmiştir.

$$\alpha = s - x_0 \tag{4.7}$$

Bu durumda, x_n değerleri $\left[\frac{x_0}{2s}, 1-\frac{x_0}{2s}\right]$ aralığında oluşur. Eğer, $\frac{x_0}{2s} = h$ olarak ifade edersek, $x_n \in [h, 1-h]$ olur. Burada x_n serisinin maksimum aralıkta düzenli bir şekilde dağılması için h parametresinin sıfıra yakın olması gerekir. Bunun için s değerinin olabildiğince büyük ve x_0 parametresinin ise olabildiğince küçük seçilmesi böyle bir dağılımı sağlayacaktır. Şekil 4.4' de, İLH sisteminden rastgele seçilmiş aynı başlangıç değeri $x_0 = 0.123456789$ ve farklı s değerleri için üretilen x_n serilerinin histogram dağılımları gösterilmiştir.



Şekil 4.4 İLH sisteminden farklı *s* değerleriyle üretilen serilerin histogram grafiği;
(a) $s = 3.2$, (b) $s = 10$, (c) $s = 99.1$, (d) $s = 500$

Histogram sonuçlarından görülebildiği gibi birbirinden çok farklı s değerleri altında bile sergilenen davranışlar birbirine çok benzerdir. Kaos dinamiğinin farklı sistem parametreleriyle benzer davranış sergilemesi böyle bir dağılımın sonucu olarak açıklanabilir. Ayrıca bu sonuçlardan anlaşılacağı üzere farklı x_n serileri, olabilecek tüm değer aralığında yayılmış ve 0.5 noktasına göre simetrik bir yapıda oluşmuştur. Sonuç olarak, yeni sistemimizde kontrol parametresi için bir üst sınır seviyesi olmadığı gibi tüm s değerleri altında İLH çıkışının benzer bir dağılım sergilemesi, şifrelemede kullanılacak uygun anahtar kodlarının oluşturulması açısından son derece önemlidir.

İLH sisteminden üretilen x_n serilerinin değer aralığı, sistemin x_0 başlangıç değeri dışında iki değişken parametreye bağlı olarak da değişmektedir. Bu durum, aynı zamanda sistemin bir önceki sisteme göre daha karmaşık bir yapıya sahip olduğunu da göstermektedir. Çünkü SLH' de sistem yörüngesi, başlangıç değeri ve kontrol parametresi olmak üzere iki parametreye bağlıydı. İLH sisteminde ise haritanın davranışı α, s ve x_0 olmak üzere toplam üç parametreye bağlıdır. Dolayısıyla sistem yörüngesinin tahmin edilebilmesi, bu parametrelerin çözülebilmesine bağlı olacaktır. Bu çalışmada, şifreleme algoritmasında kullanılacak olan anahtar kodlarının üretilmesinde İLH sisteminden faydalanılacaktır.

4.1. İyileştirilen Kaotik Haritanın İstatistiksel Analizleri

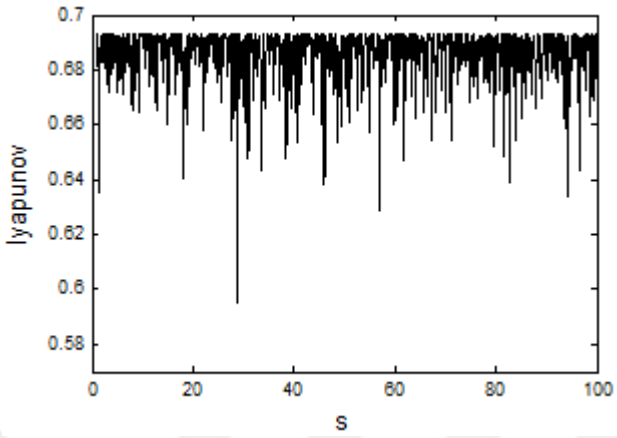
Bu kısımda, İLH sisteminin sahip olduğu dinamik davranışlar ve sistemden elde edilen serilerin karakteristik özellikleri ile ilgili analizler yapılacaktır. Sistemin sergilediği dinamik davranışın analizi için Lyapunov üsteli ile birlikte çatallaşma diyagramı incelenecek ve güç spektrum yoğunluğuna ait performansı Matlab ortamında karşılaştırmalı olarak değerlendirilecektir. Daha sonra bazı istatistiksel test yöntemleri kullanarak İLH sisteminden üretilen seriler ile ilgili rastgelelik, korelasyon gibi özellikler incelenecektir. Sonuç olarak, şifreleme kodu olarak kullanılacak olan serilerin, bu amaca yönelik uygunluğu için bir sonuca varılmaya çalışılacaktır.

4.1.1 Lyapunov Üstel Analizi

Dinamik sistemlerde kaos varlığının tespitinde bir kaç farklı yöntem bulunmaktadır. Güç spektrum analizi ve Lyapunov üstelinin hesaplanması bunlardan bazılarıdır. Bunlar arasında, Lyapunov analizi bir sistemin kaotik davranış gösterip göstermediğinin belirlenmesinde sıkça kullanılan en güçlü yöntem olup, dinamik sistemin ne kadar kaotik olduğu bu analiz yöntemiyle belirlenebilir. Kısaca, Lyapunov üsteli bir yörüngeye ait kaotik hareketin derecesini hesaplama tekniğidir [87].

Bir kaotik sistemin en belirgin özelliği başlangıç değerine olan hassas duyarlılığıdır. Lyapunov değeri bir dinamik sistemin başlangıç koşullarına olan duyarlılığının ölçülebilme kriterini ifade eder ve bu değerin pozitif olması, sistemdeki kaos varlığının kanıtıdır [35]. Kaotik bir sistemin boyut sayısı kadar Lyapunov üstelleri vardır. İLH sistemi bir boyutlu olduğundan dolayı bu sistemin sadece bir Lyapunov üsteli olacaktır.

Matlab ortamında İLH sisteminin Lyapunov katsayıları hesaplanmış ve sonuçlar Şekil 4.5' de verilmiştir.



Şekil 4.5 İLH sisteminin Lyapunov spektrumu

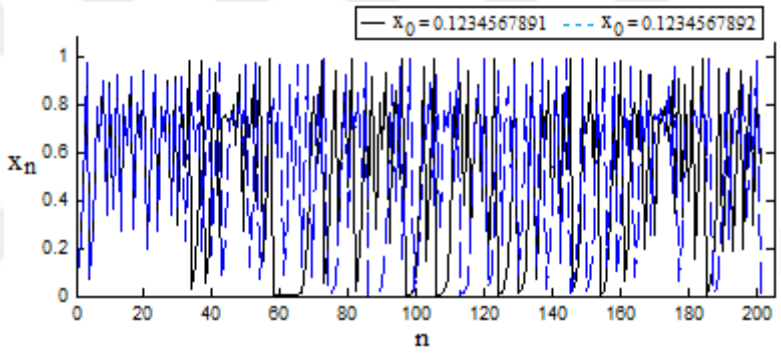
SLH sisteminde kontrol parametresi $r < 3.57$ durumunda iken kaotik olmayan davranış sergilenmektedir. Kontrol parametresi $r \in [3.57, 4]$ aralığında ve açık bölge dışında kalan alanda sistem kaotik davranış göstermekteydi. Kısaca, SLH sisteminde kontrol parametresinin tüm değerleri için sürekli bir kaos durumu mevcut değildi. İLH sistemine ait Lyapunov spektrumunda ise görüldüğü gibi tüm s değerleri için sistemin Lyapunov katsayı değerleri sürekli pozitif olup, sistem davranışı tamamen kaotiktir. Birbirine çok yakın iki başlangıç değeri ile başlatılan sisteme ait iki yörüngenin birbirinden üstel olarak uzaklaşması Lyapunov değeri ile temsil edilir. Bu durum, matematiksel olarak Denklem (4.8)' de gösterildiği gibi ifade edilebilir.

$$|\delta Z(t)| \approx e^{\lambda t} |\delta Z_0| \tag{4.8}$$

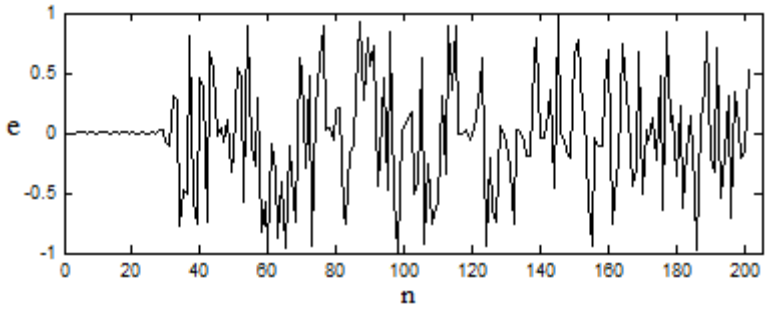
Burada, δZ_0 başlangıç değerleri arasındaki farkı gösterirken; $\delta Z(t)$ ise başlangıç değerlerine karşılık gelen sistem yörüngeleri arasındaki farkı gösterir. Dolayısıyla büyük Lyapunov değeri, birbirine çok yakın başlangıç değerleri ile elde edilmiş yörüngelerin kısa sürede birbirinden çok fazla sapacağını açıklar. Bu aynı zamanda sistem karmaşıklığının yüksek derecede olduğunu ve sistemin başlangıç değerine olan aşırı duyarlılığını da ifade eder. İLH sisteminde hesaplanan Lyapunov katsayıları, SLH sistemindeki sonuçlar ile

karşılaştırıldığında, değerlerin daha büyük veya eşit olduğu gözlemlenmiş ve buna göre iyileştirilen haritanın daha iyi kaotik özellik sergilediği sonucuna ulaşılmıştır.

İLH sisteminin x_0 ve s parametrelerine bağlı olan duyarlılığını ölçebilmek için birbirine çok yakın değerler seçilerek iki x_n serisi üretilmiştir. İlk önce $s=100$ değeri aynı kalarak, birbirine 10^{-10} kadar yakın iki başlangıç değeri sırasıyla $x_0=0.1234567891$ ve $x_0'=0.1234567892$ seçilmiş, $n=200$ iterasyon sonunda elde edilen iki yörünge Şekil 4.6(a)' da; bu yörüngelerin farkı ise Şekil 4.6(b)' de verilmiştir. Daha sonra $x_0=0.1234567891$ değeri aynı tutularak sırasıyla $s=100$ ve $s=99.9999999999$ değerleri kullanılarak benzer şekilde iki seri elde edilmiştir. Serilerin izlediği yörüngeler ve bu yörüngelerin farkı ile ilgili sonuçlar ise sırasıyla Şekil 4.7(a) ve Şekil 4.7(b)' de gösterilmiştir.



(a)

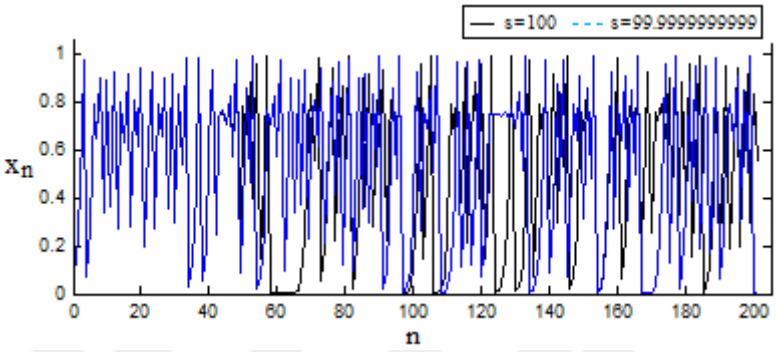


(b)

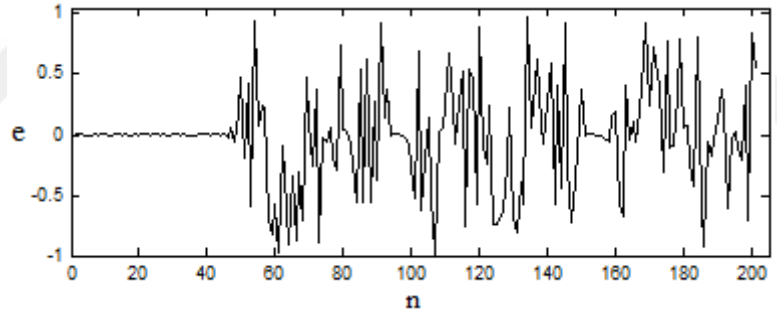
Şekil 4.6 İLH sisteminin başlangıç değerine olan duyarlılığı; (a) Birbirine çok yakın iki farklı x_0 ile üretilen yörüngeler, (b) (a)' daki yörüngeler arasındaki fark

Şekil 4.6' da verilen grafik sonucundan görüldüğü gibi İLH sistemi başlangıç değerine karşı aşırı duyarlılık sergilemektedir. Başlangıç değerinde çok ufak bir değişim ile birbirine çok yakın olarak başlayan bu iki yörünge yaklaşık olarak 30. iterasyondan sonra birbirinden ayrılarak, tamamen farklı davranışlar sergilemektedirler. Burada elde edilen

seriler arasındaki benzerlik katsayısı ise 0.0967 olarak hesaplanmıştır. Bu sonuç, iki serinin benzerlik yönünden bir ilişkisinin olmadığını da kanıtlamaktadır. Şekil 4.7' de ise sistemin s parametresine olan hassasiyeti benzer şekilde anlatılmaya çalışılmıştır. Burada ise s değerindeki çok küçük değişim ile yörüngelerin yaklaşık 45. iterasyonda birbirlerinden tamamen sapmaları açıkça görülmektedir. Bu durum, sistemin s parametresine olan hassasiyetinin bir sonucu olarak değerlendirilir.



(a)



(b)

Şekil 4.7 İLH sisteminin s parametresine olan duyarlılığı; (a) Birbirine çok yakın iki farklı s ile üretilen yörüngeler, (b) (a)' daki yörüngeler arasındaki fark

Sonuç olarak, İLH sistem parametrelerinden s ve x_0 değerlerinde 10^{-10} gibi çok küçük bir değişimde bile farklı serilerin elde edilebileceği açıkça görülebilmektedir. Bu sonuç aynı zamanda İLH sisteminde aynı olmayan sistem parametreleri ile çok fazla sayıda farklı serilerin üretilebileceğini de kanıtlar niteliktedir. Buradan çıkarılabilecek önemli bir diğer nokta, SLH' de sistem yörüngesi iki farklı parametreye x_0 ve r bağlı olarak değişirken; İLH sisteminde ise x_0, α ve s olmak üzere üç farklı parametreye bağlı olarak yörünge değişmektedir. Ayrıca, SLH sisteminde kontrol parametresi 4 ile sınırlandırılırken böyle bir durum İLH sisteminde mevcut değildir.

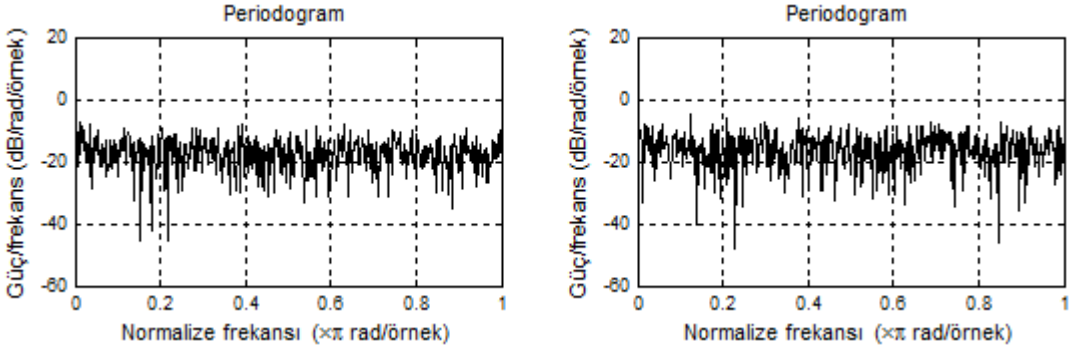
4.1.2 Güç Spektrumu Analizi

Bir sinyalin güç spektrumu, o sinyalin güç yoğunluğunun frekans bölgesindeki ölçüsüdür. Periyodik sinyaller, güç spektrumlarında temel frekans ve harmoniklerinde zirve yaparken; kaotik dinamikler ise spektrumlarında geniş bant bileşenleri verirler [88]. Nitekim bu, dinamiklerin kaotik olarak adlandırılmasında bir ölçüt olarak da kullanılabilir. Güç spektrumunun elde edilişinde en çok kullanılan yöntem, ayrık zamanlı Fourier dönüşümü ile sinyali zaman bölgesinden frekans bölgesine taşıyarak sinyal gücünün hangi frekans bileşenlerinde dağıldığını gözlemlemektir. Bu amaca yönelik olarak Periodogram, güç spektrumun hesaplanmasında kullanılan basit tekniklerden biridir ve $[x_1, x_2, ..., x_n]$ serisi olarak tanımlanmış bir sinyal için Denklem (4.9)' da verilen formül ile hesaplanır. Denklemden görüldüğü gibi Periodogram kısaca bir seri için Fourier dönüşümünün karasel büyüklüğüdür.

$$S(e^{jw}) = \frac{1}{n}\left|\sum_{l=1}^{n} x_l . e^{-jwl}\right|^2 \tag{4.9}$$

Kaotik sistemlerden üretilen seriler kaotik seri olarak adlandırılır ve bunlar rastgele serilerin önemli bir sınıfıdır. Kaosun evrensel olarak kabul görmüş bir matematiksel tanımı olmamasına rağmen kaotik seriler için genel olarak rastgele-benzeri deterministik seri tanımı kullanılır. Bu seriler $x_{n+1} = f(x_n)$ gibi bir fonksiyon kullanarak sıralı olarak elde edilir ve spektrumdaki dağılımları beyaz gürültü gibi görünür. Beyaz gürültü, birbirinden ilişkisiz rassal değerlerin oluşturduğu bir sinyal olarak tanımlanabilir. Örneğin, homojen bir dağılımın takip edildiği rastgele sayı üretecini kullanan bir beyaz gürültü sinyali Matlab ortamında '*rand*' fonksiyonu ile oluşturulabilir. Böyle bir sinyal düz bir güç spektrumuna sahiptir.

İLH sisteminde rastgele olarak seçilen $s = 10$ ve $x_0 = 0.123456789$ parametre değerleri ile bir x_n serisi üretilmiştir. Bu seriye ait Periodogram grafiği, Şekil 4.8(b)' de verilmiştir. Matlab ortamında '*rand*' fonksiyonu ile elde edilen rastgele bir serinin Periodogram grafiği ise Şekil 4.8(a)' da gösterilmiştir. Grafikteki sonuçlardan görülebildiği gibi x_n serisinin Periodogram sonucu düz bir güç spektrumuna sahiptir ve beyaz gürültününkine çok benzemektedir. Bu durum, İLH sisteminin rassal üreteç olarak kullanılmasına olanak sağlamaktadır.



(a) (b)

Şekil 4.8 Güç spektrum yoğunluğu; (a) Beyaz gürültü sinyali, (b) İLH sisteminden üretilen x_n sinyali

4.1.3 Homojenlik Analizi

İLH sisteminden elde edilen kaotik çıkış serilerinin karakteristik özellikleri açıklanan önermeler doğrultusunda ortalama değer ve öz korelasyon hesap yöntemleri kullanılarak analiz edilmiştir. Öncelikli olarak çıkış serilerinin homojen dağılımı ile ilgili merkezi eğilim ölçütü, Denklem (4.10)' da verilen ifade ile kontrol edilmiştir.

$$x_{ort} = \lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{N} x_k = 0.5 \tag{4.10}$$

Denklem (4.10)' a göre (0,1) aralığında yayılan bir kaotik serinin ortalama değeri 0.5 olmalıdır. Bu önermeye göre İLH sisteminde rastgele değerler ile seçilen parametrelerden, yarısı x_0; yarısı da s tabanlı olmak üzere toplam 100 benzetim bilgisayar ortamında gerçekleştirilmiş ve her benzetimde 100,000 seri elde edilerek ortalaması hesaplanmıştır. Benzetimler sonucunda hesaplanan bu değerlerin ortalaması x_0 için 0.4997; s için 0.5002 olarak bulunmuştur. Bu değerler, Denklem (4.10)' da verilen ideal değere çok yakındır. Bir serinin öz korelasyon değeri ise Denklem (4.11)' de gösterildiği gibi hesaplanabilir.

$$s(\beta) = \lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{N} (x_k - x_{ort})(x_{k+\beta} - x_{ort}) = 0 \tag{4.11}$$

Denklem (4.11), bir serinin kendisi ile geciktirilmiş halinin benzerliğinin karşılaştırılmasını ifade etmektedir. Rassal bir seri için bu sonuç teorik olarak 0 olmalıdır. Serideki rastgeleliği belirlemek amacıyla kullanıldığında genellikle ilk gecikme durumu, $\beta = 1$

dikkate alınır. Benzer şekilde gerçekleştirilen benzetimler sonucunda, seriler için hesaplanan ortalama öz korelasyon değeri 0.00042 olarak bulunmuştur. Bu sonuç da Denklem (4.11)' deki ideal değerine çok yakındır. Buradan ayrıca İLH sisteminden üretilen seriler için bir önceki ve bir sonraki değerleri arasında benzerlik yönünden bir ilişkinin olmadığı da anlaşılmaktadır.

Sözde-rastgele sayı üreteçlerinden, birbirleri ile bağıntı ilişkisi göstermeyen ve belirli bir aralık içerisinde eşit frekansta dağılma eğilimine sahip rassal sayılar üretmesi beklenir. İLH sistemi, tasarlanacak olan görüntü kripto sistemi için sözde-rastgele sayı üreteci olarak kullanılacağından dolayı buradan elde edilen anahtar kodlarının, iyi bir rassal sayı dizisinin sahip olduğu tüm özellikleri taşıması gerekir.

4.2. İyileştirilen Kaotik Haritanın Rastgelelik Analizleri

Rastgelelik, bir duruma ait tahmin edilebilirliğindeki eksiklik anlamına gelir ya da bir serideki düzensizliği veya anlaşılamayan uyumu öngörür. Rastgeleliğin ölçüsü ise belirsizlik veya entropi olarak nitelendirilebilir. Rastgele sayılar kriptoloji gibi birçok farklı alanda kullanılmaktadır. Bu sayıları üreten devre yapıları ise yaygın olarak kullanılan tamamen rastgele ve sözde-rastgele sayı üreteçleridir. Tamamen rastgele olan bir işlemde başlangıç durumları aynı olsa bile iki ardışık gerçekleşme birbirinden tamamen farklı olur. Dolayısıyla tamamen rastgele olan bir olay, radyoaktif bozulma veya termal gürültü gibi deterministik olmayan fiziksel olgu tabanlıdır. Bu açıdan değerlendirildiğinde, tamamen rastgele sayı üretim işlemi son derece zor ve yavaştır. Bu nedenle, tamamen rastgele sayılar yerine deterministik algoritma tabanlı olan ve rastgele benzeri anlamına gelen sözde-rastgele sayılar kullanılmaktadır. Sözde-rastgele sayı üreteçleri hiçbir ek donanıma ihtiyaç duymadan çok daha hızlı bir şekilde oluşturulabilir ve bu üreteçlerin çıkışları tamamen rastgele verilerden beklenen temel koşulları genellikle sağlarlar [89,90].

Sözde-rastgele üretilen verilerin, istatistiksel kalite doğrulaması bazı test yöntemleri ile incelenebilir. Ancak tüm test standartları, farklı bir rastgelelik özelliği değerlendirerek kendi sonuçlarını açıklayacak bir yöntem vermeye çalışır. Bu testler, bir üretecin belirli bir kriptografik uygulama için uygun olup olmadığının belirlenmesinde ilk aşama olabilir ancak hiçbir istatistiksel test, bir üretecin böyle bir uygulama için kesinlikle uygun olduğunu belirtemez. Kriptografide ise sözde-rastgele sayı üreteçlerinin amacı tahmin edilemeyen ve hesaplama ile tamamen rastgele verilerden ayırt edilemeyen değerler üretmektir.

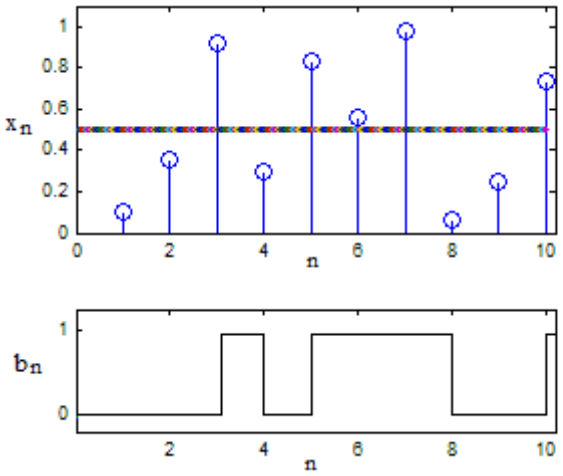
Bu kısımda, önemli bazı istatistiksel test yöntemleri kullanarak İLH çıkışının rastgelelik derecesi incelenmiştir. Bunun için, Federal Bilgi İşleme Standardı (Federal Information Processing Standards, FIPS) ve Ulusal Enstitü Standardı ve Teknolojisi (National Institute of Standards and Technology, NIST) test standartları kullanılmıştır. Eğer FIPS veya NIST standartlarına ait herhangi bir test başarısız olursa, İLH üreteci rastgelelik açısından bu standartlara göre başarısız olmuş demektir.

4.2.1. FIPS Analizi

FIPS standardında, rastgelelik derecesini belirlemek için dört istatistiksel test kullanılır ve her bir testte 20,000 bit uzunluğundaki bit serisinin durumu incelenir [91]. Ancak İLH sisteminden üretilen x_n serileri ondalıklı değerlerde olup, 0 ile 1 arasında dağılmaktadır. FIPS ve NIST testlerinin uygulanabilmesi için serinin '0' ve '1' değerlerinden oluşan bit serisi formatında olması gerekir. Bir başka ifadeyle, İLH üretecinin mevcut çıktısı, '0' ve '1' bit değerlerine haritalanması gerekir. Bu amaca yönelik olarak İLH sisteminin çıkışında uygun bir eşik seviye belirlenerek elde edilen x_n değeri, ikili sayı sisteminde tek bit ile temsil edilmiştir. İLH sisteminde simetrik bir dağılım gözlemlendiğinden dolayı bu eşik seviye 0.5 olarak seçilmiş ve sıralı bit serileri üretmek için Denklem (4.12)' de verilen dönüşüm fonksiyonu, İLH üretecinin çıkışına uygulanmıştır.

$$b_n = \begin{cases} 1 \ , \ x_n \geq 0.5 \\ 0 \ , \ x_n < 0.5 \end{cases} \tag{4.12}$$

Burada b_n, her bir x_n değerine karşılık gelen '0' veya '1' değerlerinden oluşan bit serisidir. Örneğin, $x_0 = 0.1$ başlangıç değeri ile 10 iterasyon sonunda elde edilen x_n ve ona karşılık gelen b_n grafiksel olarak Şekil 4.9' da gösterilmiştir.



x_n	b_n
0.100000	0
0.358859	0
0.919737	1
0.294076	0
0.829711	1
0.564225	1
0.982984	1
0.065471	0
0.243484	0
0.736036	1

Şekil 4.9 İLH sisteminden üretilen bit serisi

İLH sisteminde bu yöntemle üretilen bit akışlarının rastgelelik performansı FIPS ve NIST standartları altında incelenmiştir. Bu standartların uygulanmasında her bir test aşaması için Matlab ortamında rastgele olarak seçilmiş farklı x_0 ve s kullanılarak İLH üreteci üzerinden toplam 50 benzetim gerçekleştirilmiştir. Burada verilen sonuçlar, bazı testler için benzetim sonunda hesaplanan değerlerin ortalama sonuçlarıdır. Ancak tüm testler için yapılan her bir benzetime ait sayısal sonuç, analizi yapılan testin başarı şartını sağlamaktadır. FIPS standardı, bir serideki rastgelelik derecesini belirlemek için tek bit (Monobit), Poker, akış (Run) ve uzun akış testlerini kullanır. Bundan sonraki kısımda, İLH üretecinden elde edilen bit serileri bu dört teste uygulanarak her bir test için üretecin rastgelelik performansı incelenmiştir.

4.2.1.1 Tek Bit Testi

Tamamen rastgele olan bir bit serisi içerisindeki ‘0’ veya ‘1’ sayılarının olasılığı tam olarak 0.5 olması gerekir. Bu aynı zamanda, ‘0’ ve ‘1’ bit sayılarının eşit olması anlamına gelmektedir. Bu teste ilk olarak 20,000 bit serisi içindeki ‘0’ veya ‘1’ değerlerinin sayısını temsil eden x bulunur ve eğer sonuç her iki durum için $9{,}725 < x < 10{,}275$ koşulunu sağlıyorsa test başarılı olarak değerlendirilir [92].

Bilgisayar ortamında gerçekleştirilen benzetimler sonucunda, İLH sisteminden farklı başlangıç değerleriyle üretilen bit serileri içerisinde ‘1’ ve ‘0’ sayılarının ortalaması sırasıyla 0.4993 ve 0.5007 olarak bulunmuştur. Bu değerler, ‘1’ için 9,987; ‘0’ için ise 10,013 sonucuna karşılık gelmektedir. Benzer şekilde, farklı s parametre değerleri altında üretilen bit serisi oranı, ‘1’ ve ‘0’ için sırasıyla 0.4996 ve 0.5004 olarak hesaplanmıştır. Bu

sonuçlardan görüldüğü gibi her iki sistem parametresi için İLH üretecinden elde edilen farklı bit değerlerinin sayısı hemen hemen eşit sayıda olup, bunların tüm bit serisi içindeki oranı, ideal olan 0.5 değerine çok yakındır. Bu değerlendirmeye göre İLH üreteci tek bit testini başarıyla geçtiği söylenebilir.

4.2.1.2 Poker Testi

Poker testi için ilk olarak burada kullanılan 20,000 bit uzunluğundaki bit serisi 5,000 adet üst üste geçmeyen 4-bitlik kısımlara bölünür. Oluşan bu 4-bitlik örneklerin alabileceği değerlerin toplam sayısı $2^4 = 16$ olur. Daha sonra bu 4-bitlik değerler içerisinde mümkün olan her birinin oluşma sayısı bulunur ve Denklem (4.13)' de verilen formül kullanılarak bu test sonucunun değerlendirilmesinde kullanılacak olan bir *x* değeri hesaplanır.

$$x = \frac{16}{5000} \cdot \left(\sum_{i=0}^{15} f(i)^2 \right) - 5000 \tag{4.13}$$

Burada, i parametresi 4-bit ile temsil edilen değeri gösterirken; $f(i)$ ise bu değerin oluşma sayısını ifade eder. Denklem (4.13)' de bulunan sonuç eğer $2.16 < x < 46.17$ koşulunu sağlıyor ise test başarılıdır [92].

Tablo 4.1 Poker testi sonucu

Değer	Sayı
f(0)	286
f(1)	334
f(2)	315
f(3)	324
f(4)	314
f(5)	331
f(6)	337
f(7)	302
f(8)	313
f(9)	302
f(10)	326
f(11)	295
f(12)	310
f(13)	307
f(14)	287
f(15)	317

Benzetimler sonunda bit serisi içerisinde oluşturulan 4-bitlik her bir değerin oluşma frekansları Tablo 4.1' de verilmiştir. Hesaplanan sonuçlar, Denklem (4.13)' deki formüle

uygulandığında, *x* değeri 11.72 olarak bulunmuştur. Bu sonuç, bu test için kabul edilebilir aralık içerisinde olduğundan test başarılıdır.

4.2.1.3 Akış Testi

Bir bit serisi içerisinde art arda gelen maksimum sayıda '1' veya '0' değerlerinden oluşan seri bir akış olarak adlandırılır. Bu test için 20,000 bit serisindeki tüm akış sayıları bulunur ve her bir akış uzunluğu için bu test, '1' ve '0' durumlarına göre ayrı ayrı değerlendirilir. Testin başarılı olabilmesi, farklı uzunluklardaki akış sayılarının '1' ve '0' için kabul edilebilir aralıkta olmasına bağlıdır [91]. Bu test için akış sayılarına göre kabul aralığı ve test analiz sonucu Tablo 4.2' de verilmiştir.

Tablo 4.2 Akış testi sonucu

Akış uzunluğu	Kabul aralığı	1	0	Sonuç
1	$2,315 \le x \le 2,685$	2,635	2,623	Başarılı
2	$1,114 \le x \le 1,386$	1,228	1,262	Başarılı
3	$527 \le x \le 723$	655	638	Başarılı
4	$240 \le x \le 384$	322	323	Başarılı
≥ 5	$103 \le x \le 209$	144	152	Başarılı

Tablo 4.2' de verilen sonuçlara göre hem '1' hem de '0' bit değerlerine ait akış uzunlukları kabul edilebilir seviyelerde olduğundan dolayı İLH üretecinden elde edilen seriler bu teste göre rastgelelik açısından yeterli performansa sahiptir.

4.2.1.4 Uzun Akış Testi

Uzun akış testinde öncelikle 20,000 bit serisi içinde görünen sıfır veya birden oluşan en uzun akış sayısı olan *x* hesaplanır. Eğer serideki '1' ve '0' için $x < 26$ ise test başarılı olarak değerlendirilir [92]. Test sonucuna göre bit serisinde art arda gelen '1' durumu için akış sayısı 14; '0' durumu için ise 15 olarak bulunmuştur. İLH üretecinden elde edilen yeterli sayıda farklı bit serileri bu testte uygulanmış ve her durumda hesaplanan sayısal veriler testin başarıyla sonuçlandığı göstermiştir.

4.2.2. NIST Analizi

NIST standardı on beş test kullanmaktadır ve burada 1,000,000 bit uzunluğundaki seri testte uygulanır. Her bir testte, verilen bit serisi için rastgelelik varsayımı altında belirli

bir parametrenin istatistik Chi-karasel değişiminin bulunması ilk prosedür olarak benimsenmiştir. Daha sonra bulunan Chi-karasel değerden bir dönüşüm tekniğiyle rastgelelik olasılığını gösteren bir *P*-değeri hesaplanır [93]. Her bir test için hesaplanan *P*-değerinin 1' e eşit olması, teste uygulanan bit serisinin mükemmel bir rastgeleliğe sahip olduğunu; 0 değerine eşit olması ise bu serinin kesinlikle rastgele olmadığını gösterir. Dolayısıyla, NIST standardındaki her bir test [0,1] aralığında reel bir *P*-değeri üretir ve bu değer önceden tanımlı bir önem seviyesi olan α değerinden büyük ise istatistiksel test başarıyla geçilir. Genellikle α seviyesi ile ilgili olarak $0.001 \le \alpha \le 0.01$ koşulu referans alınır ve kriptografide bu seviye yaygın olarak 0.01 seçilir. Örneğin, α değerinin 0.01 olması 100 seriden birinin reddedilebileceği anlamına gelmektedir ve $P-\text{değeri} \ge 0.01$ için seri, %99 güven ile rastgele olduğu kabul edilir. NIST standardında kullanılan testler Tablo 4.3' de listelenmiştir.

Tablo 4.3 NIST testleri

Test numarası	Test adı
1	Frekans Testi (Frequency test)
2	Blok Frekans Testi (Block frequency test)
3	Akış Testi (Runs test)
4	Birler için Uzun Akış Testi (Longest-Run of ones test)
5	Matris Dizisi Testi (Matrix rank test)
6	Ayrık Fourier Dönüşümü Testi (Discrete Fourier transform test)
7	Örtüşmeyen Kalıp Eşleştirme Testi (Non-overlapping template matching test)
8	Örtüşen Kalıp Eşleştirme Testi (Overlapping template matching test)
9	Maurer Evrensel İstatistik Testi (Maurer's Universal statistical test)
10	Doğrusal Karmaşıklık Testi (Linear complexity test)
11	Seri Testi (Serial test)
12	Yaklaşık Entropi Testi (Approximate entropy test)
13	Birikmiş Toplamlar Testi (Cumulative sums test)
14	Rastgele Sapma Testi (Random excursions test)
15	Rastgele farklı Sapma Testi (Random excursions variant test)

NIST standardında Chi-karasel değerinden *P*-değerine dönüşüm için iki farklı yaklaşım benimsenmektedir. İlk yaklaşımda tek parametre olarak Chi-karasel değeri dikkate alınarak bir *P*-değeri hesaplanırken diğer yaklaşımda ise ilk parametre olarak serbestlik derecesi sayısı (*K*); ikinci parametre olarak Chi-karasel değeri kullanılarak *P*-değeri hesaplanır. Tablo 4.3' de verilen 1, 3, 6, 9, 13 ve 15 numaralı testler ilk yaklaşımı kullanırken geri kalan dokuz test ise ikinci yaklaşımı benimser. İlk yaklaşımda, Chi-karasel

parametre değerini gösteren χ^2, Denklem (4.14)' de verilen Gauss dağılım fonksiyonu yardımıyla *P*-değerine dönüşür.

$$\phi(z) = \frac{1}{\sqrt{2\pi}} . \int_{-\infty}^{z} e^{-x^2/2}\, dx \tag{4.14}$$

Gauss dağılım fonksiyonu, standart normal bir değişkenin [0,z] aralığı içindeki değerine ilişkin olasılığı verir ve burada *P*-değeri Denklem (4.15)' de verilen ifade ile bulunur.

$$P - \text{deg}eri = 1 - \phi(z) \tag{4.15}$$

Eğer Gauss dağılımı içinde *z* pozitif olması durumunda, Denklem (4.16)' da verilen hata fonksiyonu *P*-değerini hesaplamak için kullanılır.

$$erf(z) = \frac{2}{\sqrt{\pi}} . \int_{0}^{z} e^{-x^2}\, dx \tag{4.16}$$

Bu durumda, Denklem (4.17)' de tanımlanan matematiksel ifade ile *P*-değeri hesaplanmaktadır.

$$P - \text{deg}eri = 1 - erf(z) \tag{4.17}$$

Sonuç olarak, Tablo 4.3' de listelenen 1, 3, 6, 9, 15 numaralı testlerde *P*-değerinin hesaplanmasında hata fonksiyonu kullanılırken, 13 numaralı test için *P*-değeri Gauss fonksiyonu tabanlıdır. *P*-değerinin hesaplanmasına yönelik ikinci yaklaşımda ise tüm bit serisi *N* blok sayısına bölünür ve *K* serbestlik derecesiyle tüm bit serisi blok odaklı gözlemlenir. Daha sonra Chi-karasel değeri parametrenin blok odaklı teorik değerlerine karşı hesaplanır ve Denklem (4.18)' de verilen Gama fonksiyonu ile bu değere karşılık gelen bir *P*-değeri bulunur.

$$\Gamma(a, z) = \int_{0}^{z} x^{a-1} e^{-x}\, dx \tag{4.18}$$

Burada a parametresi serbestlik derecesine bağlı olup, $a = K/2$ dir ve z için ise $z = \chi^2/2$ eşitliği vardır. NIST standardına ait 2, 4, 5, 7, 8, 10, 11, 12 ve 14 numaralı testlerde *P*-değerinin bulunmasında Gama fonksiyonundan yararlanılır ve Denklem (4.19)' da verilen formül ile *P*-değeri hesaplanabilir.

$$P - \text{değeri} = 1 - \frac{r(a,x)}{r(a,\infty)} \tag{4.19}$$

NIST standardına ait bu on beş test, analiz türü açısından dört farklı kategoride ele alınabilir. NIST içinde 1-4 arasındaki testler serinin frekans karakteristiğini; 5 ve 6 numaralı testler tekrarlı yapılarını; 7-12 arasındaki testler seri ile ilgili eşleştirme ve 13-15 arasındakiler ise serinin rastgelelik özelliğini inceler.

NIST standardının uygulanmasına yönelik olarak İLH parametrelerinden $s = 999$ ve $x_0 = 0.009$ değerleri rastgele seçilerek test için yeterli uzunlukta bit serisi elde edilmiştir. Üretilen bit serisinin NIST ile ilgili test sonuçları Tablo 4.4' de verilmiştir.

Tablo 4.4 NIST analiz sonuçları

Test adı	P-değeri	Sonuç
Frekans Testi	0.9362	Başarılı
Blok Frekans Testi	0.2736	Başarılı
Akış Testi	0.1597	Başarılı
Birler için Uzun Akış Testi	0.1484	Başarılı
Matris Dizisi Testi	0.6484	Başarılı
Ayrık Fourier Dönüşümü Testi	0.3684	Başarılı
Örtüşmeyen Kalıp Eşleştirme Testi	0.9320	Başarılı
Örtüşen Kalıp Eşleştirme Testi	0.8690	Başarılı
Maurer Evrensel İstatistik Testi	0.3369	Başarılı
Doğrusal Karmaşıklık Testi	0.0513	Başarılı
Seri Testi-1 Seri Testi-2	0.9486 0.9667	Başarılı
Yaklaşık Entropi Testi	0.8972	Başarılı
Birikmiş Toplamlar Testi (ileri) Birikmiş Toplamlar Testi (geri)	0.5753 0.6476	Başarılı
Rastgele Sapma Testi	0.2995	Başarılı
Rastgele farklı Sapma Testi	0.6508	Başarılı

FIPS ve NIST test sonuçlarına göre İLH sisteminden elde edilen bit serilerinin rastgeleliği bu standartlar altında doğrulanmıştır. Dolayısıyla İLH üreteci, FIPS ve NIST standartlarına göre rastgelelik anlamında başarılı olduğu ve yeterli performansı sergilediği söylenebilir.

4.3. Serilerin Öz Korelasyon Analizi

Korelasyon iki değişken arasındaki ilişki derecesini belirten istatistiksel bir tekniktir. Öz ve çapraz korelasyon fonksiyonları, serilerin rastgelelik durumları ile ilgili bilgi veren iki önemli ölçümdür. Olasılık ve istatistikte, çapraz korelasyon terimi iki rastgele değişken arasındaki ilişkiyi belirlemek için kullanılır. Bir sinyalin kendisiyle çapraz korelasyonu ise sinyalin öz korelasyonu olarak tanımlanmaktadır. Öz korelasyon fonksiyonu, bir sinyalin kendisi ile gecikmiş versiyonu arasındaki ilişkiyi ölçerek, bu gözlemler arasındaki benzerliği belirler. Ayrık zamanlı iki rastgele sinyal için d gecikmesindeki r çapraz korelasyonu Denklem (4.20)' deki gibi tanımlanmaktadır.

$$r_d = \frac{\sum_i \left[(x[i] - \bar{x}).(y[i-d] - \bar{y}) \right]}{\sqrt{\sum_i (x[i] - \bar{x})^2} \sqrt{\sum_i (y[i-d] - \bar{y})^2}} \tag{4.20}$$

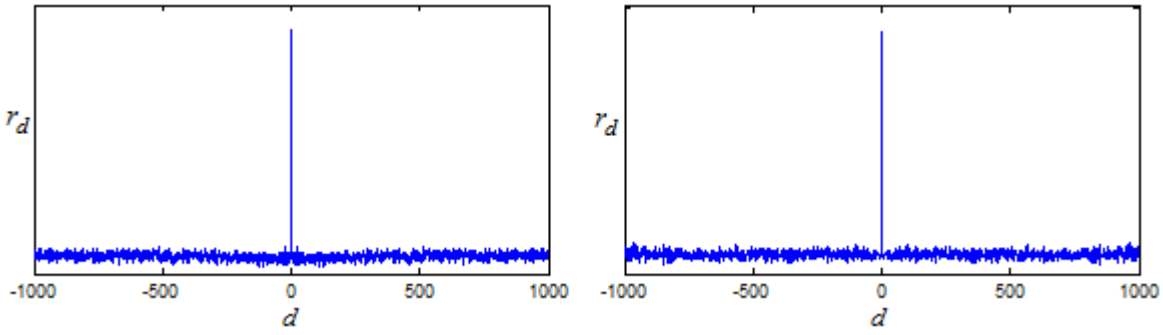
Burada $\bar{x}$, $\bar{y}$ sırasıyla x ve y serilerinin ortalama değerlerini gösterirken, korelasyon katsayısı olan r_d ise bu iki seri arasındaki doğrusal ilişkinin büyüklük ölçüsünü ifade eder. Denklem (4.20)' de bulunan paydadaki ifade, korelasyon katsayı değer aralığını [-1,1] sınırında tutacak şekilde normalleştirir. Eğer x ve y serileri arasında pozitif güçlü bir korelasyon var ise r_d değeri, 1' e yakın; negatif güçlü bir korelasyon var ise -1' e yakın ve çok zayıf bir korelasyon durumunda ise 0' a yakın olacaktır. Bu iki seri arasında herhangi bir doğrusal ilişki yok ise o zaman korelasyon katsayısı 0 olacaktır.

Öz korelasyon analizi ile bir sinyal içindeki tekrarlı yapı veya herhangi bir periyodik durum tespit edilebilir. Dolayısıyla bir seriye ait rastgelelik durumunun belirlenmesinde bu analiz bir ölçüt olarak kullanılabilir. Rastgele bir sinyalin öz korelasyon analizinde, bu sinyalin geçmiş ve gelecek durumlarının birbiriyle ilişkisiz olması ve sadece mevcut durumunda güçlü bir bağıntının oluşması beklenir. Böyle bir sinyalin sahip olduğu bu karakteristik özellik aynı zamanda sinyal içerisinde kendini tekrar eden bir uyumun olmadığını da gösterir. Buna karşın periyodik bir sinyal için gecikme, sinyal periyoduna eşit veya tam bir katı olduğu zamanlarda bu sinyal, geciktirilmiş haliyle mükemmel bir korelasyon sağlayacak ve sinyalin öz korelasyon fonksiyonu bu gecikme değerlerinde ani sıçrama yapacaktır. Diğer bir yandan tamamen rastgele özelliğe sahip bir sinyalin öz korelasyonu ise bir dürtü fonksiyonu gibi oluşacaktır.

Yeterince uzun bir $x[n]$ serisi için öz korelasyon fonksiyonu, Denklem (4.20)' ye göre uyarlandığında eşitlikteki payda yaklaşık bir ifadeyle basitleştirilebilir ve seriye ait farklı gecikme zamanlarındaki öz korelasyon katsayısı Denklem (4.21)' deki gibi hesaplanabilir. Öz korelasyon fonksiyonunun, gecikme zamanını gösteren eksene göre grafiksel gösterimi korelogram olarak adlandırılmaktadır.

$$r_d = \frac{\sum_{i=d+1}^{N} \left[(x[i]-\bar{x}).(x[i-d]-\bar{x})\right]}{\sum_{i=1}^{N}(x[i]-\bar{x})^2} \tag{4.21}$$

Bu kısımda, öz korelasyon analiziyle İLH sisteminden elde edilen serilerin farklı gecikme aralığındaki korelasyon durumları incelenecektir. Daha açık bir ifadeyle, serilerin kendisi ile zaman gecikmeli durumunun benzerlik yönünden karşılaştırılması ele alınacaktır. Böylelikle seri içeresinde tekrarlı bir uyumun olup olmadığı görülebilecektir. İlk olarak İLH sisteminden 10,000 iterasyon ile yeterli uzunlukta bir çıkış serisi elde edilmiş ve serinin öz korelasyon katsayıları hesaplanmıştır. Bu seriye ait korelogram grafiği Şekil 4.10(a)' da gösterilmiştir. Daha sonra Matlab ortamında '*rand*' fonksiyonu kullanılarak aynı büyüklükte bir rastgele seri dizisi oluşturulmuştur. Bu serinin korelogram grafiği ise Şekil 4.10(b)' de verilmiştir.

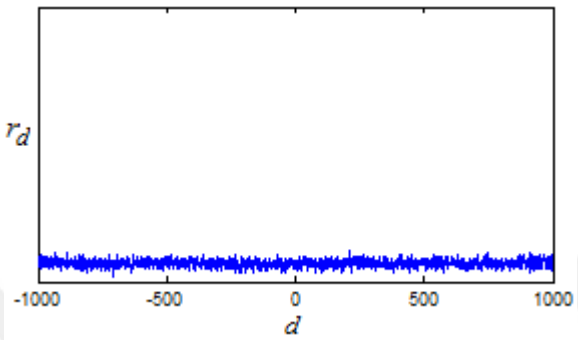


(a) (b)

Şekil 4.10 Korelogram analizi; (a) İLH üretecinden elde edilmiş seri, (b) Matlab ortamında '*rand*' fonksiyonu ile oluşturulmuş seri

Geniş aralıkta rastgele bir serinin $d \neq 0$ gecikmesindeki öz korelasyon fonksiyonun büyüklüğü, $d = 0$ gecikmesindeki değerinden küçük yani $r_{xx}(0) \geq r_{xx}(d)$ olmalıdır. Şekil 4.10(a)' da gözlemlendiği gibi İLH üretecinden elde edilen serinin öz korelasyonu, sadece

$d = 0$ durumunda zirve yapmış ve bir dürtü fonksiyonu gibidir. Bu özellik aynı zamanda rassal bir serinin bu analiz altındaki davranış karakteristiğidir. Matlab ortamında üretilen serinin karakteristiği de Şekil 4.10(b)' de görüldüğü gibi rassal bir seri ile benzerlik göstermektedir. Şekil 4.11' de ise bu iki serinin çapraz korelasyonu grafiksel olarak verilmiştir.



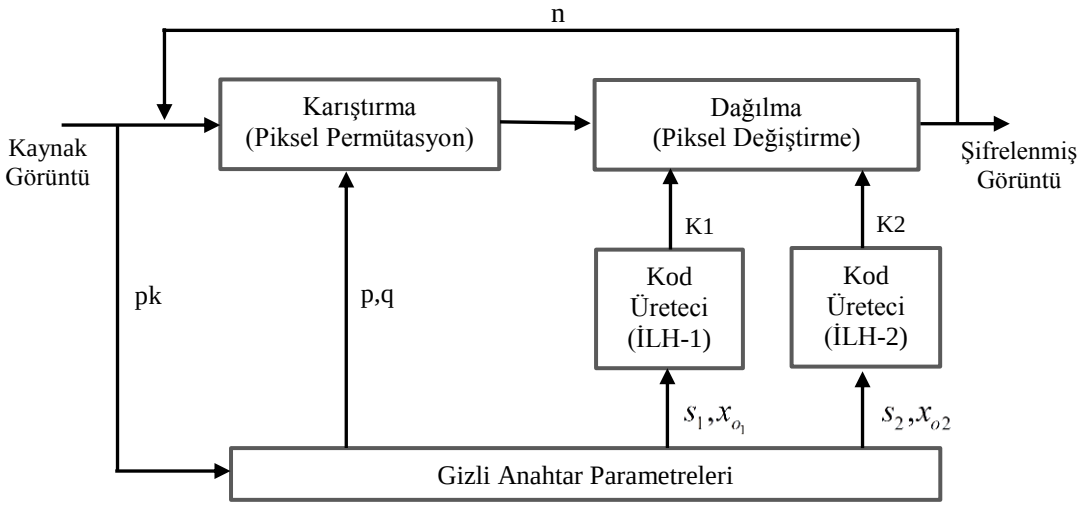
Şekil 4.11 İLH serisi ile Matlab ortamında üretilmiş serinin çapraz korelasyonu

Sonuç olarak İLH üretecinden elde edilen serilerin öz korelasyon analiz sonucuna göre serilerde periyodiklik gibi tekrarlı bir uyumun olmadığı ayrıca serilerin kendileri ile geçmiş durumlarının benzerlik yönünden birbirleriyle hiçbir ilişki göstermediği gözlemlenmiştir.

5. SUNULAN KAOS TABANLI GÖRÜNTÜ ŞİFRELEME ALGORİTMASI

Bu kısımda, görüntü şifrelenmesinde kullanılacak olan şifreleme ve çözme algoritmaları detaylı bir şekilde anlatılacaktır. Kaynak görüntüye bağlı olmasıyla şifreleme anahtarlarını doğrudan etkileyecek bir mekanizma yapısını içermesi, bu çalışmada sunulan kriptografik algoritmanın özgün tarafı olacaktır. Bu mekanizmanın tasarımında, görüntü içerisindeki herhangi bir pikselde bir bit değişimde bile tamamen farklı şifreleme kodlarının üretilmesi amaçlanmıştır. Şifreleme algoritmasının görüntü bilgisine olan bağlılığı ile kripto sistemin diferansiyel saldırılara karşı daha güçlü olması planlanmıştır. Ayrıca şifreleme işlemi, birkaç kez tekrarlanarak gerçekleştirilecek ve her bir iterasyonda farklı şifreleme kodları kullanılacaktır. Böylelikle kripto sistemin daha karmaşık ve rastgele bir çıkış üretmesi hedeflenmiştir. Kripto sistemin anahtar aralığının arttırılmasında da bu özelliklerin katkısı olacaktır.

Kaos tabanlı görüntü şifreleme ilkesinin temelinde, kaotik seriler kullanarak orijinal bilgi üzerinde işlem gerektirmesi ve oluşan şifrelenmiş bilginin gürültü benzeri rastgele bir duruma dönüşmesi yatar. Burada kullanılan kaotik anahtar serileri, kaos oluşturan sistemin kontrol parametresine ve gizli anahtar olarak düşünülen başlangıç değerine bağlıdır. Kaos tabanlı görüntü şifreleme yapıları genellikle iki kısımdan oluşur. Şifreleme düzeninin ilk basamağı olan karıştırma işleminde pikseller kendi aralarında yer değiştirerek karışmış görüntüyü oluşturmakta ve daha sonra bu karışmış görüntünün piksel değerleri sırayla değiştirilerek şifreleme tamamlanmaktadır. İyi bir şifreleme için gerekli olan bu iki kısım, burada sunulan şifreleme yapısında da gerçekleştirilmiştir. Sunulan simetrik tabanlı görüntü şifreleme algoritmasının blok şeması Şekil 5.1’ de gösterilmiştir.

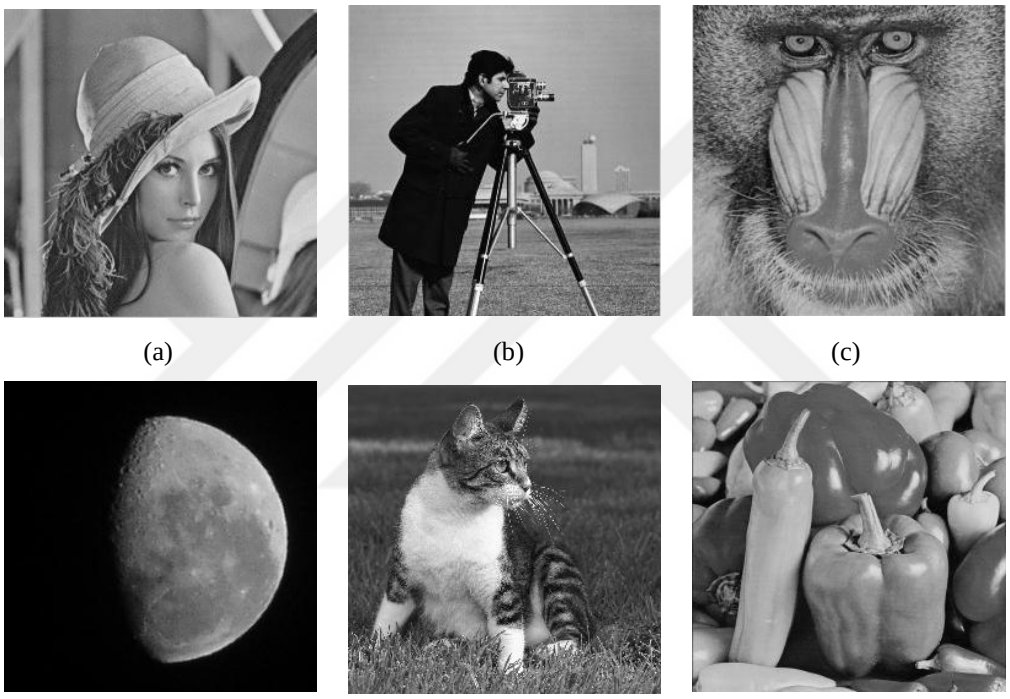


Şekil 5.1 Sunulan görüntü şifreleme algoritmasının blok şeması

Bu çalışmada şifreleme işleminde kullanılan gizli kodlar, daha önceden tasarlanmış ve istatistiksel analizleri başarıyla gerçekleştirilmiş İLH sisteminden elde edilmiştir. Sunulan kriptografik yapıda şifreleme algoritmasında kullanılmak üzere iki farklı anahtar grubu (K1 ve K2) için iki İLH sistemi üreteç olarak kullanılmıştır. Dağılma işleminde iki farklı anahtar grubunu kullanmadaki amaç, kripto sistemin anahtar aralığını arttırarak özellikle kaba-kuvvet saldırısına karşı sistemi daha güçlü bir yapıya büründürmektir. Bu özellik, aynı zamanda seçilen-bilgi saldırı tekniğine karşı da sistemi daha dayanıklı yapacaktır. Sunulan kripto sistemde esas şifreleme dağılma kısmında gerçekleşmektedir. Çalışma içerisinde kullanılan görüntü şifreleme kavramı, görüntüyü oluşturan tüm piksellerin değerce değişmesi demektir. Dolayısıyla şifrelenmiş görüntü ile kaynak görüntü arasında ölçü olarak bir fark olmayacak ancak donanım ortamında bellekte kapladığı alanlar farklı olabilecektir.

Sunulan kripto sistem, kaynak girdisi olarak renkli veya gri görüntü kullanabilir. Renkli görüntüler bilgisayar ekranlarında 24-bit derinliğinde veri olarak görüntülenir. Görüntüleme, **R** (Kırmızı), **G** (Yeşil) ve **B** (Mavi) olmak üzere aynı görüntünün üç katmanından oluşur ve bu katmanların üst üste binmesiyle ekrana yansır. Dolayısıyla RGB görüntüdeki her bir piksel bu üç rengin kombinasyonu ile temsil edilir. Gri görüntü ise tek katmandan oluşur ve 8-bit derinliğe sahiptir. Bir renkli görüntü aslında farklı renk tonları ile temsil edilen üç gri görüntüden oluşur. Bu anlamda, kripto sistemde aynı görüntünün gri ölçekli katmanlarının ayrı ayrı şifreleme işleminde kullanılması mümkündür. Kripto sistem ile ilgili olarak bundan sonraki teorik ve bilgisayar ortamındaki deneysel

çalışmalarda kaynak girdisi olarak gri görüntüler; çalışmanın uygulama tarafında ise renkli görüntüler kullanılmıştır. Sunulan şifreleme algoritmasının hız, performans ve güvenlik analizlerinde elliden fazla test görüntüsü kullanılmıştır. Algoritmaya ait istatistiksel ve diferansiyel analizler bilgisayar ortamında Matlab yazılımı kullanılarak gerçekleştirilmiştir. Şifrelenmiş görüntüler ile ilgili bazı istatistiksel analizler için ortalama değerler hesaplanmıştır. Kripto sistemde en sık kullanılan bazı test görüntüler Şekil 5.2' de gösterilmiştir.



(a) (b) (c)

(d) (e) (f)

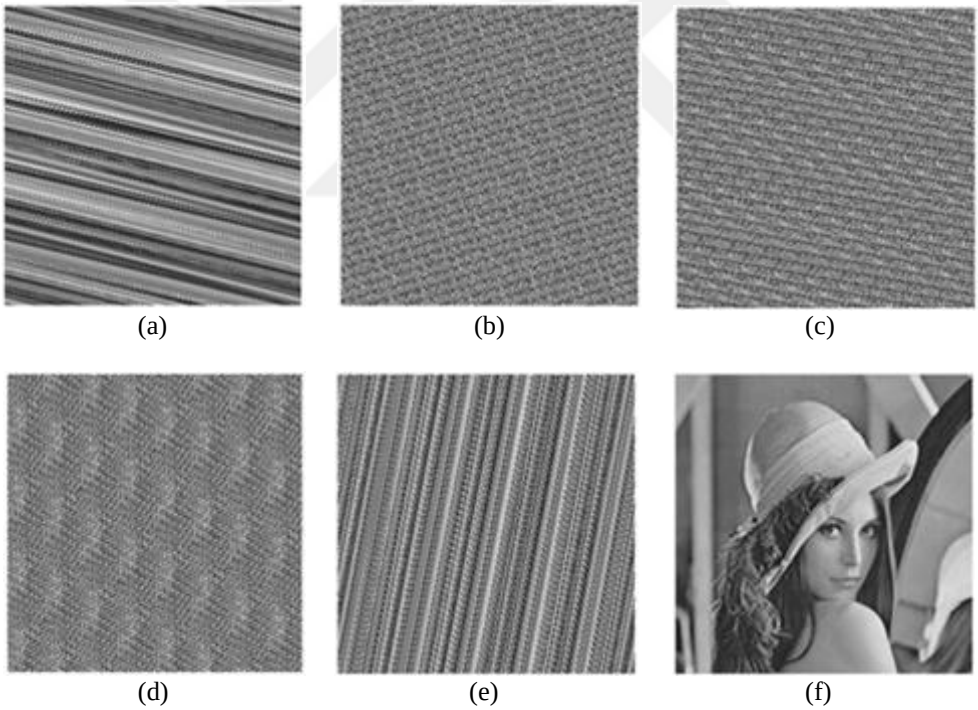
Şekil 5.2 Kripto sistemde sıkça kullanılan test görüntüleri; (a) Lena, (b) Kameraman, (c) Babun, (d) Ay, (e) Kedi, (f) Biber

5.1. Sunulan Algoritmanın Karıştırma Kısmı

Kripto sistemin ilk aşaması olan karıştırma kısmında, kaynak görüntünün piksel pozisyonlarının değişimi için AKH yöntemi kullanılmıştır. Burada AKH yönteminin tercih edilmesinde, basit bir modüler işlemin dışında sadece doğrusal bir dönüşümün gerçekleşmesi etken olmuştur. Ayrıca AKH yönteminin yeterli sayıdaki iterasyon sonucuyla piksellerin ilk konumlarına geri dönmesi, çözme işleminde tekrar orijinal görüntünün elde edilmesi noktasında önemlidir. Karıştırma işlemi sonucunda oluşan karışmış görüntüdeki piksellerin yer değişimi, AKH' a uygulanan iterasyon sayısı olan *n*

dışında p ve q parametrelerine de bağlıdır. Bu parametreler, kripto sistemin karıştırma kısmındaki şifreleme anahtarlarını ifade ederken aynı zamanda sistemimize fazladan iki gizli anahtar da sağlamaktadır.

Anlamlı bir görüntüde yakın komşu pikselleri birbirleri ile güçlü bir bağıntı içerisindedirler ve aralarındaki ilişki katsayı değeri büyüktür. Dolayısıyla, görüntü kripto sisteminin şifreleme kalitesini arttırmak için esas şifreleme işleminden önce bu güçlü bağıntının kırılması gerekir. Karıştırma, bu amaca yönelik olarak gerçekleştirilen bir permütasyon işlemidir ve görüntü içerisindeki her bir piksel pozisyonunu rastgele bir şekilde yer değiştirme amacı taşır. Örneğin, AKH parametreleri rastgele bir şekilde $p = 5$ ve $q = 4$ gibi seçilsin. Daha sonra AKH yöntemi, 256×256 ölçüsündeki 'Lena' test görüntüsüne farklı iterasyon sayıları altında uygulansın. Uygulama sonucu oluşan karışmış görüntülere ait sonuçlar Şekil 5.3' de gösterilmiştir.



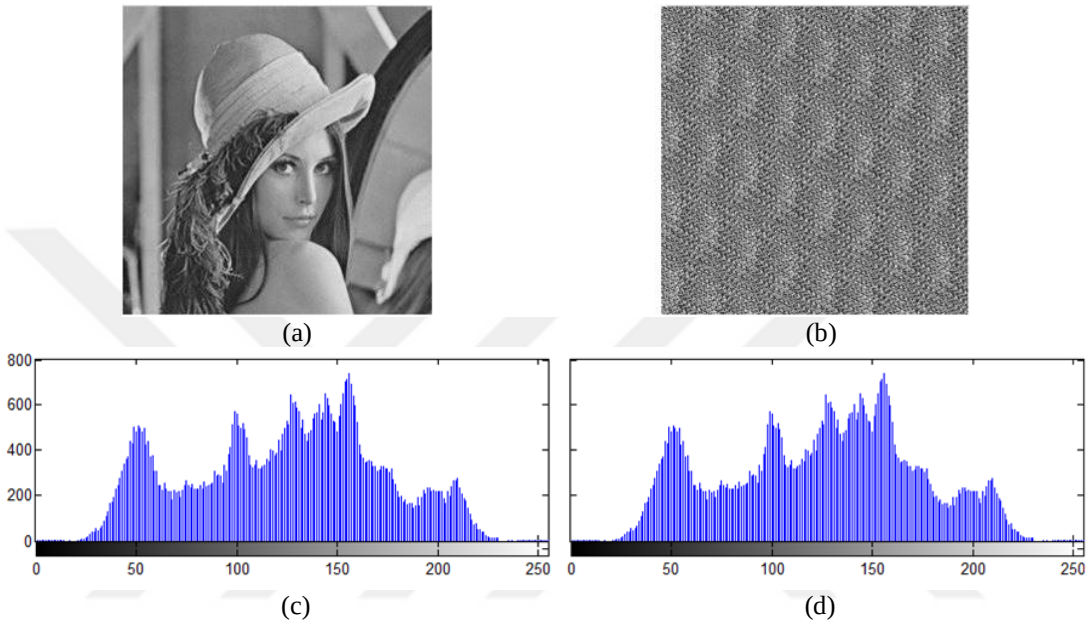
(a) (b) (c)

(d) (e) (f)

Şekil 5.3 'Lena' görüntüsüne AKH yönteminin uygulanışı; (a) $n = 1$, (b) $n = 3$, (c) $n = 7$, (d) $n = 16$, (e) $n = 511$, (f) $n = 512$

Şekil 5.3' deki sonuçlara göre, AKH parametrelerinden $p = 5$ ve $q = 4$ değerleri için 'Lena' görüntüsünün periyodu 512 olmaktadır. Yani orijinal 'Lena' görüntüsüne ait piksel pozisyonlarının tamamı 512 iterasyon sonunda bulunduğu ilk yerlerine dönüş yapmaktadır. Ancak bu işlem sonunda kaynak görüntüye ait piksellerin değerlerinde bir değişim olmaz

sadece pozisyonları etkilenir. Dolayısıyla karışmış görüntü ile orijinal görüntünün histogram dağılımları aynı olacaktır. Görüntü işlemede histogram, bir görüntü içerisindeki piksel değerlerinin dağılımının temsilinde kullanılan istatistiksel bir analizdir. 'Lena' görüntüsünün ve bu görüntünün AKH sonucu elde edilen karışmış halinin histogram grafiği Şekil 5.4' de verilmiştir.



Şekil 5.4 (a) Orijinal 'Lena' görüntüsü, (b) Karışmış 'Lena' görüntüsü, (c) 'Lena' görüntüsünün histogram grafiği, (d) Karışmış 'Lena' görüntüsünün histogram grafiği

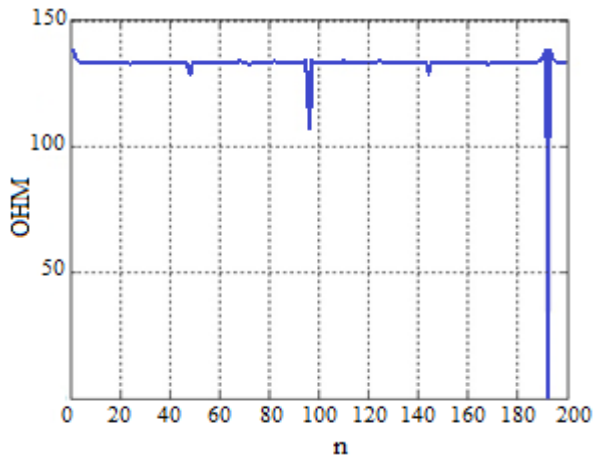
Şekil 5.4' de verilen histogram grafiklerinden görüldüğü gibi karışmış ve orijinal görüntülerin histogram dağılımları tamamen aynıdır ve bu sonuç iterasyon sayısından bağımsızdır. Dolayısıyla görüntünün düzgün bir şekilde bulanıklaşması, AKH sonucu piksellerin kendi aralarında yer değiştirmesi ile açıklanmaktadır. Ancak bu özellik, sadece karıştırma fonksiyonu içeren sistemleri istatistiksel saldırılara karşı açık hale getirir ve sistem güvenliği açısından bu yöntemin tekrarlanabilir olmasından dolayı AKH tek başına yeterli güvenlik sağlayamaz. Sistem güvenliğini arttırmak için bir aşamaya daha gerek vardır. Karıştırma sonucu görüntü piksellerinin birbirinden ne derece uzaklaştığı ve pikseller arasındaki güçlü bağıntının ne derecede kırıldığı incelenmesi gereken bir diğer önemli konudur.

5.1.1 Karışmış Görüntüde Ortalama Hareket Mesafesi

Görüntü karıştırma, anlamlı bir görüntünün düzensiz ve belirsiz hale getirilerek gerçek anlamının gizlenmesine yönelik gerçekleştirilen bir işlemdir. Gizli bir karıştırma işlemi, seçilmiş görüntü saldırılarına karşı hesaplama karmaşıklığını arttırarak şifreleme algoritmasının kripto analizini daha karmaşık bir yapıya sokar. Bu işlem sonucu görüntü içerisindeki yakın pikseller birbirinden ne kadar çok uzaklaşırsa aralarındaki güçlü bağıntının kırılması o kadar etkili olur. Bu dereceyi ölçmek için karışmış görüntüye ait ortalama hareket mesafesi (OHM) analizi kullanılmıştır. Bu değer, orijinal görüntü ile karışmış görüntünün piksel pozisyonlarının birbirinden ortalama ne kadar uzaklaştığını ifade eder ve Denklem (5.1)' deki gibi tanımlanır.

$$OHM = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \sqrt{(w-i)^2 + (v-j)^2} \tag{5.1}$$

Burada (i, j) ve (w,v) sırasıyla orijinal ve karışmış görüntülerdeki piksel koordinatlarını; (M,N) ise görüntü ölçüsünü göstermektedir. Denklem (5.1)' de OHM değerinin büyük olması, orijinal görüntü ile karışmış görüntü arasında zayıf bir ilişkinin olduğunu ve dolayısıyla karıştırma verimliliğinin yüksek olduğunu belirtir [94]. Bu çalışmada gerçekleştirilen OHM analizi için 256×256 ölçüsündeki 'Lena' test görüntüsü örnek olarak kullanılmıştır. Bu analizde, $p = q = 1$ değerleri seçilmiş ve analiz sonuçları grafiksel olarak Şekil 5.5' de verilmiştir.



Şekil 5.5 'Lena' görüntüsü için OHM analizi

Şekil 5.5' de verilen grafiksel sonuçtan görüldüğü gibi belirli bir iterasyon değerinde OHM değeri 0 olmuştur. Bu durum, AKH sonucu karışmış görüntüdeki piksel pozisyonlarının, orijinal görüntüdeki ilk konumlarına geri döndüğünü yani kaynak görüntünün hiç karışmadığını göstermektedir. Bu noktadaki iterasyon değeri aynı zamanda kaynak görüntünün periyodu şeklinde yorumlanabilir. Burada 'Lena' görüntüsünün periyodunun 192 olduğu açıktır. Şekil 5.3' de gösterilen aynı ölçüdeki 'Lena' görüntüsünün periyodunun 512 olması, AKH parametrelerinden p ve q değerlerinin burada farklı olmasından gelmektedir.

256×256 ölçüsüne sahip bir görüntü için rastgele karıştırma algoritmasında hesaplanmış ortalama evrensel değer 85.33 dür [94]. Bu çalışmada ise hesaplanan en yüksek OHM değeri 134 seviyesindedir. Bu sonuç, evrensel ortalamaya göre gayet iyi bir değerdir ve karıştırma işleminin yüksek verimliliğe sahip olduğunu göstermektedir. Ancak karıştırma verimliliğini etkileyen bir başka faktör vardır. Karıştırma işleminde piksellerin yer değiştirmesi sonucunda belirli bir pozisyona gelen yeni piksel ile o pozisyondan çıkan piksel değeri aynı olabilir. Bir görüntüde komşu piksellerin değerleri birbirine yakın olduğu düşünülürse belli bir koordinattaki yeni ve eski piksel değerlerinin karıştırma sonucunda aynı olması istenmeyen bir durum olacaktır. Dolayısıyla görüntü karıştırma işlemi sonunda aynı koordinattaki orijinal ve karışmış görüntüdeki piksel değerlerinin karşılaştırması ile ilgili analiz bir sonraki kısımda incelenmiştir.

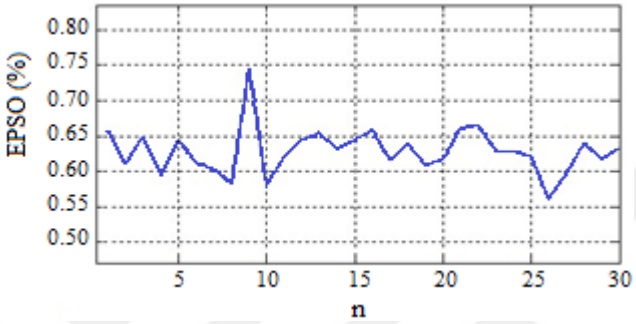
5.1.2 Karışmış Görüntüde Eşit Piksel Sayılarının Oranı

Eşit Piksel Sayılarının oranı (EPSO), karışmış görüntü ile orijinal görüntü arasındaki belirli koordinatlarda bulunan aynı piksel değerlerinin sayısıyla ilgili bir orandır. Bu parametre, orijinal ve karışmış görüntülerin aynı pozisyondaki piksel değerlerinin hangi oranda birbirinden farklı olduğunu gösterir ve Denklem (5.2)' de verildiği gibi hesaplanır.

$$EPSO(R,U)=\frac{1}{M\times N}\sum_{i=1}^{M}\sum_{j=1}^{N}[T(R(i,j),U(i,j))]\times100\,\% \tag{5.2}$$

$$T(R(i,j),U(i,j))=\begin{cases}1 & , \quad R(i,j)=U(i,j)\\ 0 & , \quad R(i,j)\neq U(i,j)\end{cases} \tag{5.3}$$

Denklem (5.3)' de, R ve U sırasıyla orijinal ve karışmış görüntüyü temsil ederken; T ise belli koordinattaki eşit piksellerin sayısını göstermektedir. EPSO analizinde T değerinin sıfıra eşit olması karışmış görüntü içerisindeki piksellerin, orijinal görüntüsündeki aynı koordinata karşılık gelen piksellerden tamamen farklı olduğunu açıklar. Şekil 5.6' da verilen grafik, 'Lena' test görüntüsü için farklı iterasyon değerlerindeki EPSO sonuçlarını göstermektedir.



Şekil 5.6 'Lena' görüntüsü için EPSO analizi

Grafiksel sonuç incelendiğinde, EPSO analizi için en iyi ve en düşük değer % 0.56 ile 26. iterasyonda gerçekleşmiştir. Bu sonuç, yeri değiştirilen 10,000 pikselden sadece 56 tanesinin pozisyonu değişen piksel ile aynı olduğunu ifade etmektedir. Benzer şekilde farklı büyüklükteki test görüntüleri (Çiçek, Kameraman, Babun ve Manzara) $p = 5$ ve $q = 4$ parametre değerleriyle karıştırma işleminde kullanılmış ve ilk beş iterasyon için EPSO sonuçları Tablo 5.1' de verilmiştir.

Tablo 5.1 Farklı test görüntüleri için EPSO analiz sonuçları

iterasyon	**Test görüntüleri**			
	Çiçek (128x128)	**Kameraman** (256x256)	**Babun** (512x512)	**Manzara** (1024x1024)
n	EPSO (%)	EPSO (%)	EPSO (%)	EPSO (%)
1	0.67	1	0.74	0.72
2	0.64	0.98	0.77	0.71
3	0.73	0.98	0.73	0.73
4	0.65	0.99	0.78	0.73
5	0.68	0.95	0.77	0.72

5.1.3 Karışmış Görüntüde Pikseller Arasındaki İlişki

Anlamlı bir görüntüde komşu pikseller arasındaki ilişki katsayısı büyüktür. Bu güçlü ilişkinin esas şifrelemeden önce kırılması veya zayıflatılması gerekir. Kripto sistemin karıştırma kısmında kullanılan AKH yönteminin sisteme olan etkisini ölçmek için pikseller arasında benzerlik analizi yapılmıştır. Bu analizde, köşegen yönündeki pikseller referans alınmış ve piksellerin yer değişimi sonrasındaki ilişki katsayıları Denklem (5.4)' de gösterilen formül ile hesaplanmıştır.

$$cc = \frac{\sum_{i=1}^{N}(x_i - \bar{x}).(y_i - \bar{y})}{\sqrt{(\sum_{i=1}^{N}(x_i - \bar{x})^2).(\sum_{i=1}^{N}(y_i - \bar{y})^2)}} \tag{5.4}$$

Burada x, y köşegen yöndeki komşu pikselleri gösterirken, $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$ ve $\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$ ortalama değerleri temsil etmektedir. N ise çapraz pozisyondaki (x, y) çiftinin sayısını ifade eder. Tablo 5.2, bazı test görüntülerine ait piksellerin AKH yöntemi uygulanmadan önceki benzerlik katsayı değerlerini göstermektedir.

Tablo 5.2 AKH öncesi piksellerin benzerlik katsayıları

Test görüntüleri			
Çiçek (128x128)	**Kameraman** (256x256)	**Babun** (512x512)	**Manzara** (1024x1024)
0.89561	0.89237	0.95401	0.97320

Tablo 5.2' de verilen sonuçlarda beklenildiği gibi pikseller arasındaki benzerlik katsayıları tam bir uyuşmayı gösteren 1 değerine çok yakındır. Daha sonra aynı test görüntüleri AKH yöntemiyle karıştırılmıştır. Oluşan karışmış görüntülere ait piksellerin benzerlik katsayıları hesaplanmış ve sonuçlar Tablo 5.3' de verilmiştir.

Tablo 5.3 AKH sonrası piksellerin benzerlik katsayıları

n	Test görüntüleri			
	Çiçek (128x128)	**Kameraman** (256x256)	**Babun** (512x512)	**Manzara** (1024x1024)
1	0.22224	0.65950	0.60295	0.70364
2	0.07556	0.01021	-0.16899	-0.01078
3	0.13694	-0.01862	0.10817	-0.16973
4	-0.19015	0.12543	0.03703	-0.23430
5	-0.19356	-0.03351	0.00744	-0.17756

Bulunan sonuçlar, karıştırma sonrasında görüntüye ait komşu pikseller arasındaki benzerlik ilişkisinin önemli ölçüde azaldığını göstermektedir. Özellikle birinci iterasyondan sonra görüntü pikselleri ile ilgili katsayı değerlerinde ciddi azalmalar gerçekleşmiştir. Burada negatif katsayı değeri, iki değişkenden birinde ki artışın diğerinde azalma ile ilişkili olduğu anlamına gelmektedir. Sonuç itibariyle AKH yönteminin piksellerin yer değiştirmesinde etkili olduğu söylenebilir. Ancak bu karışmış görüntülere ait hesaplanan benzerlik katsayı değerleri, tatmin edici bir seviyede değildir. Çünkü burada sadece pozisyona dayalı bir permütasyon işlemi gerçekleşmiştir. Esas şifrelemenin yani piksel değerleri ile ilgili değişimin yapılacağı dağılma kısmında benzerlik katsayılarının daha da düşmesi muhtemel bir sonuçtur. Bu anlamda, kripto sistemde dağılma işleminin gerekliliği önem arz eder.

5.2. Sunulan Algoritmanın Dağılma Kısmı

İlk aşamada gerçekleştirilen permütasyon işlemiyle pikseller arasındaki ilişki katsayıları zayıflık anlamında iyileştirilmişti. Ancak bu iyileştirme piksellere ait sadece pozisyon tabanlı olduğundan burada piksel değerleriyle ilgili bir istatistiksel farklılık oluşmamıştı. Dolayısıyla aynı histogram grafiğine sahip olan karışmış görüntü, orijinal görüntü hakkında yararlı bilgiler içermeye devam eder. Bu durum, sadece permütasyon işlemi kullanan sistemlerin kriptografik anlamda güvenli olmadığını gösterir. Sonuç itibariyle, daha güvenli bir kriptografik yapı için ek bir işleme daha ihtiyaç olduğu açıktır.

Sunulan kripto sistemde esas şifreleme, karışmış görüntüye ait piksellerin tamamının değerce değiştiği dağılma kısmında gerçekleşmektedir. Dağılma kısmı, kripto sistemin piksel permütasyon işleminden sonra uygulanan ikinci ve son aşamasıdır. Karıştırma ve dağılma sınıfındaki görüntü kripto sistemlerinin en zahmetli kısmı dağılma işlemidir çünkü hesaplama yükünün önemli bir kısmı burada gerçekleşir. Kripto sistemin bu kısmında piksel değerlerinin değiştirilmesinde özgün bir dağılma fonksiyonu kullanılmıştır. Bu

fonksiyon, matematiksel işlemlerden modüler aritmetik ve mantıksal işlem içermektedir. Kripto sistemde kullanılan dağılma kısmı ile ilgili detaylara geçmeden önce İLH sisteminden şifreleme kodlarının nasıl üretildiği, öncelikle ele alınması gereken bir konudur.

Gri bir görüntüde her bir piksel 8-bit ile temsil edilir. Bu değer, ondalık tabanda 0 ile 255 arasında bir sayıya denk gelir. Burada 0 genellikle siyah renk kodu ile gösterilirken; 255 ise beyaz rengi ifade eder. Dağılma fonksiyonun uygulanabilmesi için şifreleme kodu ile piksel değeri aynı formatta olmak zorundadır. Ancak daha önceden belirtildiği gibi kaotik İLH sistemi 0 ile 1 arasında ondalıklı değerler üretmektedir. Dolayısıyla bu değerlerin belirli bir dönüşüm fonksiyonuyla 0 ile 255 arasında sayılara dönüştürülmesi, şifreleme algoritmasında kullanılan matematiksel fonksiyonların gerçekleştirilebilmesi açısından gereklidir. Dönüştürülen bu değerler, kripto sistemin dağılma kısmında kullanılacak olan gizli şifreleme kodları olacaktır. Şifreleme kodlarının elde edilmesine yönelik olarak Denklem (5.5)' de verilen dönüşüm fonksiyonu İLH üretecinin çıkışına uygulanmıştır.

$$kod = \mathrm{mod}(round(x_n \times 10^9), 256) \tag{5.5}$$

Burada, *round()* fonksiyonu hesaplanan değeri en yakın tamsayıya yuvarlamak; *mod()* fonksiyonu ise anahtar değerini 0 ile 255 arasında sınırlandırmak için kullanılmıştır. Ayrıca İLH çıkışının 10^9 gibi büyük bir değer ile çarpılması, üretilen kod değerindeki hassasiyeti arttırma amacına yönelik olarak düşünülmüş bir işlemdir. Karışmış görüntüdeki piksel değerlerini şifreleme kodlarıyla karıştırarak, şifreli ve kaynak görüntü arasındaki ilişkiyi anlamsızlaştırmak dağılma kısmının temel amacıdır. Bunun sonucunda da, şifrelenmiş görüntüde kaynak görüntüsündekinden tamamen farklı bir histogram dağılımının oluşması beklenir. Sunulan kripto sistemde piksel değerlerinin değişimi için Denklem (5.6)' da verilen algoritma dağılma fonksiyonu olarak kullanılmıştır.

$$c(i) = k_1(i) \oplus \left\{ p(i) - c(i-1) + k_2(i)^2 + 255 \right\} \bmod 256 \tag{5.6}$$

Burada, *p(i)*, *c(i)* ve *c(i-1)* sırasıyla karışmış görüntü pikselini; şifrelenmiş pikseli ve bir önceki şifreli pikseli temsil etmektedir. Bunun dışında $k_1(i)$ ve $k_2(i)$ olmak üzere iki ayrı

şifreleme kodu, dağılma fonksiyonu içerisinde karışmış görüntü pikselini değiştirmek için kullanılmıştır. Denklem (5.6)' da bulunan " $p(i)-c(i-1)+k_2(i)^2$ " işleminin negatif olma ihtimaline karşın ilgili terime 255 ilave edilerek sonucun sıfır veya pozitif olması sağlanmıştır. Dağılma fonksiyonunda gerçekleştirilen tüm işlemler, yazılım ve donanım ortamında yüksek hızlarda yapılabildiğinden dolayı böyle bir algoritma verimli bir şifreleme sunacaktır.

Verilen dağılma fonksiyonunda dikkat edilirse mevcut şifrelenmiş piksel, bir önceki şifreli piksel değerine bağlıdır. Bu nedenle, kaynak görüntüdeki küçük bir değişim şifrelenmiş görüntü üzerinde birden fazla pikseli etkileyecek ve bu da dağılımın tüm şifreli görüntüye yansımasını sağlayacaktır. Bu özellik, diferansiyel saldırılara karşı kripto sistemi daha güçlü yapacaktır. Burada, karışmış görüntüye ait tüm pikseller şifreleme işlemi tamamlanıncaya kadar sırayla şifrelenmektedir. Ancak böyle bir şifreli seri başlatmak için $c(0)$ olarak tanımlanan bir başlangıç değeri gereklidir. Bu değer, kripto sistemde önceden tanımlanabildiği gibi şifreleme parametrelerine bağlı olarak da belirlenebilir. Temel şifreleme algoritması için bu ilk değer, Denklem (5.7)' deki gibi belirlenmiştir.

$$c(0) = \mathrm{mod}(round(\alpha_1 + \alpha_2 + pk + n) \times 10^5), 256) \tag{5.7}$$

Burada α, İLH sistemiyle ilişkili bir parametredir ve çalışmanın dördüncü bölümünde anlatıldığı gibi sisteminin kontrol parametresi ile başlangıç değeri arasındaki farkı gösterir. Kripto sistem, kontrol ve başlangıç değerleri sırayla s_1, s_2 ve x_0, x_0' olan iki farklı İLH sistemini üreteç olarak kullanmaktadır. Bu gizli parametreler aynı zamanda kripto sistemin dağılma kısmına ait şifreleme anahtarlarıdır. İLH sistem parametreleri ile α arasındaki ilişki Denklem (5.8)' de belirtilmiştir.

$$\begin{aligned} \alpha_1 &= s_1 - x_0 \\ \alpha_2 &= s_2 - x_0' \end{aligned} \tag{5.8}$$

Denklem (5.6)' da verilen şifrelemenin ilk başlangıç değeri $c(0)$ bir kaynak görüntüsünün karşılığı olan şifreli görüntüsünün bir pikseli olmadığı gibi karışmış görüntüye ait tüm şifreli değerleri tamamen etkileyebilecek bir parametre kaynağıdır. Bu açıdan bakıldığında

bu değer, İLH sisteminin parametreleri dışında *pk* ve *n* iterasyon sayısına bağlı olacak şekilde tasarlanmıştır. Dolayısıyla algoritmada kullanılan bu ilk başlangıç değeri, kripto sistemin tüm anahtar parametrelerine bağlı olması itibariyle sunulan kripto sistem, şifreleme anahtarlara karşı hassas duyarlılık sağlayacaktır. Ancak güvenli bir kripto sistem şifreleme anahtarları dışında aynı zamanda kaynak görüntüye de bağlı olması gerekir. Yani kripto sisteme uygulanan kaynak görüntü değiştiği anda algoritmada kullanılan şifreleme kodlarıyla beraber sistem çıkışının da değişmesi beklenir. Bu anlamda tasarlanacak kripto sistem, şifrelenecek olan kaynak görüntüye karşı da hassas duyarlılık gösterecektir.

5.2.1 Algoritmanın Kaynak Görüntüye Bağlılığı

Kaynak görüntü ile bağıntılı ve şifreleme anahtarlarını etkileyebilecek son derece hassas bir değerin bulunabilmesi, şifrelemede önemli derecede etkili olacaktır. Böyle bir algoritmanın aynı zamanda diferansiyel saldırılara karşı da dayanıklı olduğu mevcut bazı çalışmalarla kanıtlanmıştır [95-101]. Ancak burada kaynak görüntü ile ilişkili parametrenin bazı temel özellikleri sağlaması gerekir. Bu anlamda, ilgili parametre öncelikli olarak kaynak görüntü hakkında fazla bilgi vermemesi gerekir. Çünkü aksi durumda, saldırgan bu değeri analiz ederek şifreli görüntüyü kırabilir. Ayrıca bu değer, kaynak görüntü ile güçlü bir bağıntı sağlaması gerekir. Yani kaynak görüntüde ufak bir değişimde bile bu değerin büyük oranda değişmesi gerekir. Bu özellik, şifreli görüntünün diferansiyel saldırılara karşı korunmasında tek yöntemdir [102].

Sunulan kripto sistemde kaynak görüntüye olan hassas bağlılık, algoritma tasarımında dikkate alınmıştır. Bu hassasiyetteki amaç, kaynak görüntüdeki çok ufak bir değişimin, buna karşılık gelen şifrelenmiş görüntüsünü tamamen değiştirmesidir. Küçük bir piksel değerinin değişiminde bile tamamen farklı şifreleme kodlarının üretilmesi ve algoritmada kullanılması, bu dinamiğin altında yatan gerçektir. Dolayısıyla, algoritmada kullanılan şifreleme kodları, şifrelenecek görüntüye göre değişecektir. Daha açık bir ifadeyle, aynı şifreleme anahtarları farklı görüntüler için kullanılmayacaktır. Kripto sistemde bu özellik, kaynak görüntünün karakteristiği ile ilişkili bir *pk* parametresinin algoritmaya eklenmesi ile sağlanmıştır. Bu parametre, Denklem (5.9)' da verilen ifade ile tanımlanmıştır.

$$pk = \left| \sum_{i=1}^{N \times N} (-1)^i \{normalize\_görüntü\} \right| + \max(normalize\_görüntü) + \frac{1}{N}$$
$$normalize\_görüntü = \frac{kaynak\_görüntü}{256} \tag{5.9}$$

Burada şifrelenecek olan karasel görüntünün ölçüsü *N* ile gösterilirken, normalize edilmiş görüntü ise $1 \times N^2$ ölçülerindedir. Görüntünün piksel değerlerine ve ölçüsüne bağlı olarak istatistiksel bir sayısal değer veren *pk,* sunulan kripto sistemde bir şifreleme anahtarı gibi değerlendirilmemiştir. Bu parametre İLH üretecindeki başlangıç değerini güncelleyerek her bir şifreleme iterasyonunda farklı kodlar üretmeyi sağlayan bir değişken parametre olarak kullanılmıştır. Dolayısıyla algoritma içerisinde her bir iterasyonda farklı şifreleme kodlarının oluşumunda etkin bir rol oynamaktadır. İLH üretecinin başlangıç değerinin, *pk* parametresi ile değiştirilmesine yönelik ifade Denklem (5.10)' da verilmiştir.

$$X_0 = \mathrm{mod}((x_0 + pk), 1) \tag{5.10}$$

Denklem (5.9)' a göre *pk* değeri matematiksel olarak sıfır olamaz. Çünkü bu parametre kaynak görüntünün ölçüsüne de bağlıdır ve sürekli olarak pozitiftir. Dolayısıyla kripto sistemde anahtar üretecinin başlangıç değerinin her bir iterasyonda kesinlikle değişeceği açıktır. Bazı test görüntülerine ait *pk* değerleri Tablo 5.4' de verilmiştir.

Tablo 5.4 Farklı ölçülerdeki bazı test görüntülerin *pk* değerleri

Test Görüntüleri	**Lena** (128x128)	**Kameraman** (256x256)	**Babun** (512x512)	**Kedi** (1024x1024)	**Kaplan** (2048x2048)
pk	7.355468	21.707031	63.216796	95.172851	483.723144

Algoritmaya eklenen ve kaynak görüntüye bağlı olan bu yapı, kripto sistemin tüm parametreleri aynı olsa bile şifrelenecek görüntüde bir bitlik değişimde dahi üreteç sisteminden tamamen farklı anahtar kodlarının üretilmesini sağlayacaktır. Örneğin, Matlab ortamında görüntünün işlenmesi sonrasında 'Kameraman' test görüntüsüne ait P(1,1) pozisyonundaki piksel değerinin 157 olduğu görülmüştür. Bu değerde sadece bir bit değişim uygulanarak yeni değer 158 yapılmış ve daha sonra her iki gri görüntü için aynı s ve x_0 parametreleri altında şifreleme kodları elde edilmiştir. Her iki durum için hesaplanan

pk değerleri ve üretilen kodların bazıları Tablo 5.5' de verilmiştir. Tablo sonucundan da görülebildiği gibi aynı sistem parametreleri kullanılmasına rağmen oluşan şifreleme kodları birbirinden tamamen farklıdır.

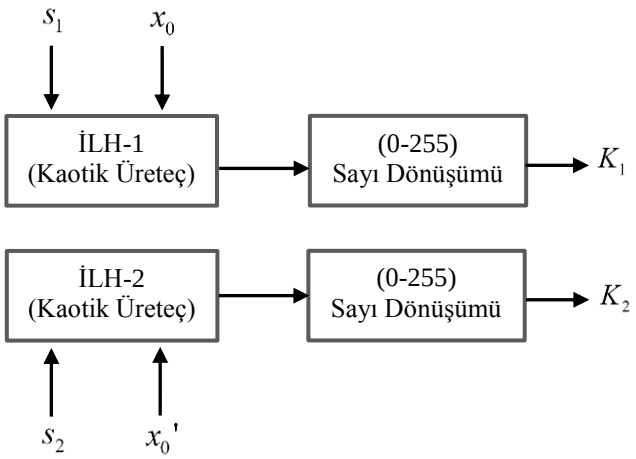
Tablo 5.5 Aynı sistem parametreleriyle farklı anahtar kodların üretimi

Kod-1 P(1,1)=157 pk=21.707031	**Kod-2** P(1,1)=158 pk=21.703125
132	239
19	64
11	221
199	129
248	197
57	226
210	45
80	90

5.3. Şifreleme Kodları ve İstatistiksel Analizler

Bir kripto sistemin en önemli kısmı bilgi şifrelemede kullanılan anahtarlardır. Kripto sistem içerisinde kullanılan şifreleme kodlarının rastgeleliği, oluşan şifrelenmiş bilgideki rastgeleliği doğrudan etkileyecektir. Bu anlamda anahtar serisinin rastgeleliği, bir şifreleyicinin en önemli hassas içeriğidir ve sistem güvenliğini doğrudan etkiler. Ayrıca kullanılan anahtar kodlarının belirli bir bölgede eşit dağılım göstermesi yani tekdüzelik özelliğine sahip olması, iyi bir şifreleme anahtarından beklenen özelliktir.

Sunulan kripto sistemde piksel değerinin şifrelenmesinde, K_1 ve K_2 olmak üzere iki farklı şifreleme kodu aynı dağılma fonksiyonu içerisinde kullanılmıştır. İLH üretecinde şifreleme kodlarının üretilmesiyle ilgili blok şema Şekil 5.7' de gösterilmiştir. Bu kısımda, K_1 ve K_2 anahtar kodlarının her iterasyonda nasıl değişkenlik göstereceği açıklanmış ve bunların tekdüzelik ve entropi analizleri yapılmıştır.



Şekil 5.7 K_1 ve K_2 şifreleme kodlarının üretimi

İLH-1 ve İLH-2 üreteçlerinden sırasıyla K_1 ve K_2 şifreleme kodları önce Denklem (5.5)' e göre elde edilip, daha sonra Denklem (5.6)' da verilen dağılma fonksiyonunda kullanılmaktadır. Burada her bir iterasyonda kullanılan K_1 ve K_2 şifreleme kodlarının uzunluğu, kaynak görüntüdeki piksel sayısı kadardır. Bunun yanında, K_1 ve K_2 arasındaki ilişki, kripto sistemin algoritmik yapısı ile ilgili kritik bir noktayı temsil etmektedir. Örneğin, $K_1 \neq K_2$ olması, tüm kripto sistemin daha büyük bir anahtar alanına sahip olmasında önemlidir. Şifreleme anahtarlarındaki bu farklılık, bu çalışmada üreteçlere ait kontrol parametrelerinin veya başlangıç değerlerinin her bir iterasyonda değiştirilmesiyle gerçekleştirilmiştir.

Sunulan şifreleme algoritmasında, K_1 şifreleme kodunun elde edildiği İLH-1 üretecinin başlangıç değeri x_0, her bir şifreleme iterasyonu sonunda şifrelenmiş görüntü bilgisine ait yeni bir *pk* değeri ile Denklem (5.10)' a göre güncellenerek bir sonraki iterasyon için tamamen farklı bir şifreleme anahtarı oluşturulmaktadır. Farklı K_2 kodunun üretilmesi ise İLH-2 üretecine ait s_2 değerinin her bir iterasyon sonunda kendi başlangıç değeri, x_0' kadar arttırılması sonucu oluşan yeni bir s_2 değerinin, İLH-2 üretecinde kullanılmasıyla gerçekleştirilmektedir. Parametredeki bu değişikliği özetleyen ifade Denklem (5.11)' de verilmiştir.

$$S_2 = s_2 + x_0' \tag{5.11}$$

Kısaca, sunulan kripto sistemde İLH-1 ve İLH-2 üreteçlerinin sırasıyla başlangıç değeri ve kontrol parametresi her bir şifreleme sonunda değiştirilerek, farklı K_1 ve K_2 şifreleme kodları elde edilmektedir. Bundan sonraki kısımda ise üretilen şifreleme kodları ile ilgili sıkça kullanılan istatistiksel analizlerden tekdüzelik, hassasiyetlik ve belirsizlik durumları incelenmiştir.

5.3.1 Tekdüzelik Analizi

Kripto sistemde kullanılan gizli anahtar kodlarıyla ilgili olarak tekdüzelik analizi Chi-karasel (Chi-square) test yöntemi kullanılarak değerlendirilmiştir. Chi-karasel testi, herhangi bir dağılımı çözümlemek için beklenilen ideal değerler ile gözlemlenen değerlerin karşılaştırılmasına dayalı bir yöntem olup, Denklem (5.12)' deki gibi tanımlanır.

$$\chi^2 = \sum_{i=1}^{n} \frac{(o_i - e_i)^2}{e_i} \tag{5.12}$$

Burada o_i gözlemlenen frekansı; e_i beklenilen ideal frekansı ve χ^2 ise Chi-karasel değerini gösterir. Denklem (5.12)' den açıkça görüldüğü gibi herhangi bir dağılım içinde gözlemlenen frekans ideal frekansa yakın ise χ^2 değeri sıfıra yakın küçük bir değer olacaktır. Gözlemlenen frekans ile ideal frekans değerleri arasında önemli bir farkın olması halinde ise χ^2 değeri büyük olacaktır. Eğer gözlemlenen ve beklenilen frekanslar eşit olduğunda $\chi^2 = 0$ olacaktır. İyi bir tekdüzelik dağılım için serideki her bir elemanın homojen olarak dağılması yani birbirleri ile eşit frekansta gözlemlenmesi gerekir. Tekdüzelik bir dağılımda her bir eleman için beklenilen frekans Denklem (5.13)' deki gibi bulunur.

$$e_i = \frac{N}{n} \tag{5.13}$$

Burada *n* eleman sayısını gösterirken; *N* ise toplam gözlem sayısını temsil eder. Örneğin, her bir piksel değerinin 8-bit yani onluk sayı sisteminde 0 ile 255 arasında olduğu 256×256 ölçüsündeki bir gri görüntü düşünüldüğünde, bu görüntü için her bir piksel değeri bir elemanı temsil ederken, tüm piksellerin sayısı ise toplam gözlem sayısını belirtir.

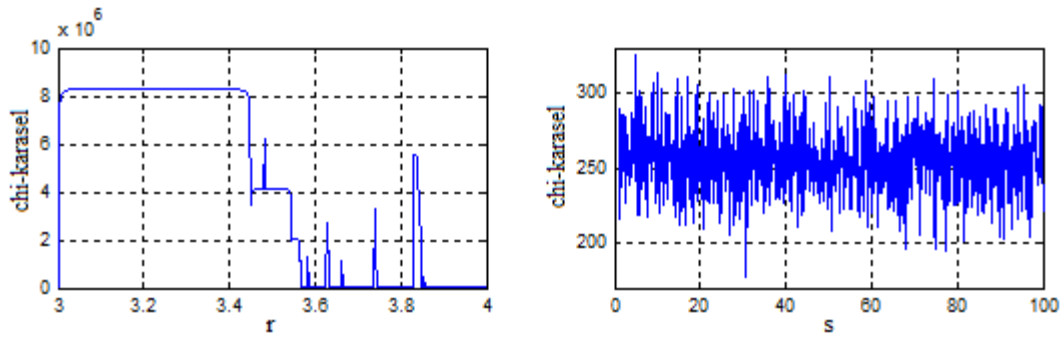
Dolayısıyla, tam bir tekdüzelik dağılımında her bir piksel için beklenilen ideal frekans, Denklem (5.14)' de hesaplandığı gibi olur.

$$e_i = \frac{N}{n} = \frac{256 \times 256}{256} = 256 \tag{5.14}$$

Bulunan bu değer, tam tekdüzelik dağılım için tüm piksel değerlerinin birbirine eşit sayıda yani 256 olması gerektiğini açıklamaktadır. Sonuç olarak, böyle bir görüntüde tekdüzelik analizi için Chi-karasel test fonksiyonu Denklem (5.15)' deki gibi tanımlanmaktadır.

$$\chi^2 = \sum_{i=0}^{255} \frac{(o_i - 256)^2}{256} \tag{5.15}$$

Bu çalışmada Chi-karasel test fonksiyonu, İLH sisteminden üretilen anahtar serilerinin dağılımı ile istenilen homojen dağılım arasındaki ilişkiyi ölçmek için kullanılmıştır. Bununla birlikte İLH sisteminde tekdüzelik performansının daha iyi değerlendirilebilmesi amacıyla üretecin farklı kontrol parametreleri altında Chi-karasel değerlerinin hesaplanması ve SLH sistemindeki sonuçlar ile karşılaştırılması gerekir. Buna göre her iki sistem için Chi-karasel sonuçları, farklı parametre değerleri altında hesaplanmış ve grafiksel olarak Şekil 5.8' de gösterilmiştir.



(a) (b)

Şekil 5.8 Chi-karasel test sonuçları; (a) SLH sistemi, (b) İLH sistemi

Şekil 5.8(a)' da verilen grafiksel sonuçtan görüldüğü gibi SLH sisteminin, periyodik veya sabit davranışın sergilendiği yani kaotik olmadığı durumunda hesaplanan Chi-karasel sonuçları 10^6 seviyelerinde olup, bunlar tekdüzelik kriterine göre çok büyük değerlerdir. Bu sonuçlar, sistem çıktısının bu şartlar altında tekdüzelik bir dağılıma sahip olmadığını

göstermektedir. Bununla birlikte, sistemin sadece kaotik davranış sergilediği kontrol parametresinin belli bazı değerlerinde Chi-karasel değerleri kabul edilebilir seviyelerdedir. Eleman aralığı 256 olan durumlar için Chi-karasel değeri en fazla 293 olabilir [103]. Dolayısıyla, bu seviyenin altındaki tüm değerler kabul edilebilir ve dağılımın iyi bir tekdüzeliğe sahip olduğunu belirtir. İLH sistemine ait Şekil 5.8(b)' de verilen test sonuçlarına göre tüm s parametreleri için Chi-karasel sonuçları yeterince küçük değerlerde olup, sistem çıktısı tamamen tekdüzelik özelliği göstermektedir. İLH sistemi için hesaplanan Chi-karasel değerler, grafik sonucundan da görülebildiği gibi ortalama 250 seviyesindedir. Sonuç olarak her iki sistemin tekdüzelik performansları birbirleri ile karşılaştırıldığında, İLH sisteminin çok daha iyi bir seviyede olduğu açıkça görülmektedir. Tablo 5.6' da ise aynı başlangıç değeri, $x_0 = 0.123456789$ için SLH ve İLH sistemlerinin farklı kontrol parametreleri altındaki bazı Chi-karasel sonuçları verilmiştir.

Tablo 5.6 SLH ve İLH sistemlerinde Chi-karasel sonuçların karşılaştırılması

SLH	χ^2	**İLH**	χ^2
r = 2.4	1.6×10^7	s = 3.2	**249**
r = 3.2	8.3×10^6	s = 4	**240**
r = 3.83	5.5×10^6	s = 49	**255**
r = 3.9	293	s = 87	**234**
r = 4	277	s = 100	**225**

5.3.2 Hassasiyetlik Analizi

Şifrelemede kullanılan kodların sistem parametrelerine karşı hassas duyarlı olması gerekir. Bu özellik, anahtarda gerçekleşen çok küçük bir değişim sonucunda tamamen farklı bir şifreli bilgi oluşturacaktır. Algoritmada kullanılan şifreleme kodlarının kripto sistem parametrelerine olan hassasiyetini ölçmek için İLH üretecinden birbirine çok yakın parametre değerleri ile 100,000 iterasyon gerçekleştirilmiştir. İlk 10 iterasyon için hesaplanan anahtar kodları Tablo 5.7' de verilmiştir.

Tablo 5.7 Birbirine çok yakın iki farklı sistem parametresiyle üretilen kodlar

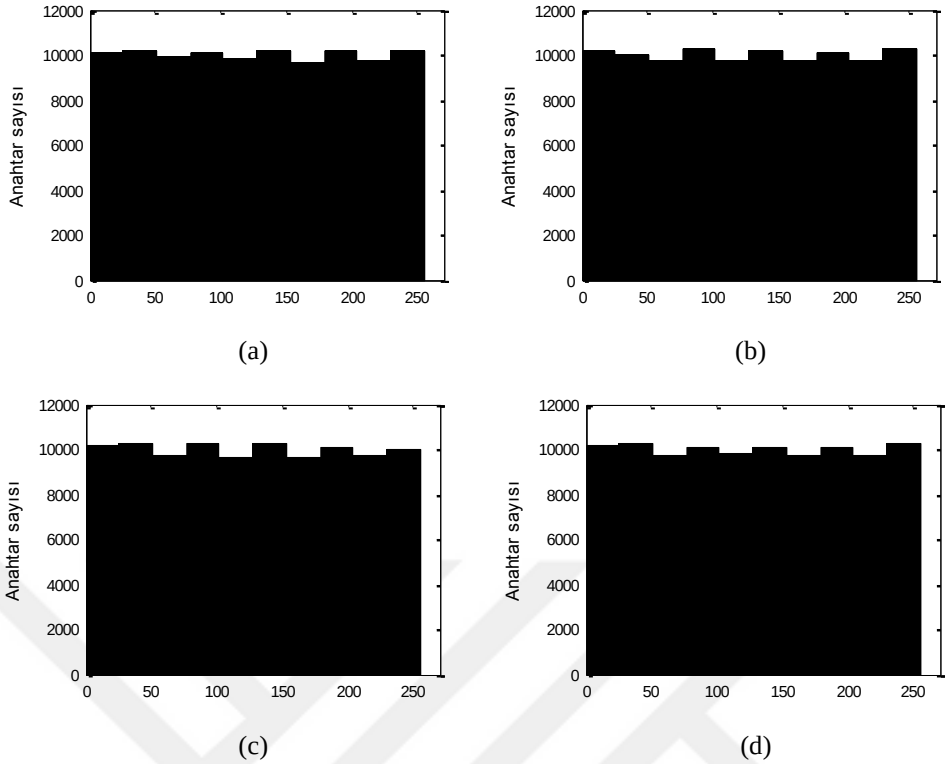
$s = 99$ $x_0 = 0.987654321$ K_1	$s = 99$ $x_0 = 0.987654322$ K_2	$s = 98.99999$ $x_0 = 0.987654321$ K_1`	$s = 98.99999$ $x_0 = 0.987654322$ K_2`
246	242	244	240
173	158	166	151
201	155	178	132
59	24	41	6
66	195	130	3
149	11	77	195
228	50	10	89
226	169	72	15
207	7	115	170
51	188	147	29

Bu sonuçlardan kolayca görülebildiği gibi başlangıç veya kontrol parametrelerinde çok küçük bir değişimde bile üretilen anahtar kodları birbirinden tamamen farklıdır. Bu durum, aynı zamanda İLH üretecinden elde edilen değerlerin şifreleme parametrelerine olan hassas duyarlılığını koruduğunu göstermektedir. Tablo 5.8' de ise bu anahtar serileri arasındaki ilişkiyi ifade eden benzerlik katsayıları verilmiştir. Bu sonuçlara göre üretilen anahtar kodları arasında benzerlik yönünden bir ilişkinin olmadığı açıktır.

Tablo 5.8 Şifreleme kodları arasındaki benzerlik katsayıları

	K_1`	K_2`
K_1	0.00308	0.00467
K_2	-0.00021	-0.00244

Algoritmada kullanılan anahtar kodlarından farklı sistem parametreleri altında homojen olarak dağılması beklenir. Bu açıdan bir değerlendirme yapılırsa, birbirinden çok farklı sistem parametreleriyle elde edilen şifreleme kodlarının dağılımları, histogram grafikleri üzerinden incelenmiştir. Analiz sonunda elde edilen grafiksel sonuçlar Şekil 5.9' da verilmiştir.



Şekil 5.9 Farklı parametreler ile elde edilen serilerin histogramları; (a) $s = 99$ ve $x_0 = 0.123$ (b) $s = 25$ ve $x_0 = 0.734$, (c) $s = 10.688$ ve $x_0 = 0.0193$, (d) $s = 247.3$ ve $x_0 = 0.99$

Rastgele seçilmiş farklı parametreler ile üretilen anahtar kodlarının histogram grafiği hemen hemen düz bir dağılıma yani tekdüze olduğu ve üretilen sayıların oluşma frekanslarının birbirine çok yakın olduğu grafiksel sonuçlardan görülebilmektedir. Gerçekleşen 65536 iterasyon için K_1 ve K_2 dağılımları içinde rastgele seçilmiş bazı sayıların oluşma frekansları Tablo 5.9' da verilmiştir.

Tablo 5.9 Dağılım içerisindeki bazı kodların oluşma sayısı

K_1		K_2	
Kod	**Frekans**	**Kod**	**Frekans**
0	268	9	254
17	238	28	266
62	245	113	284
169	257	206	258
255	262	250	271

5.3.3 Belirsizlik Analizi

Rastgele sayı üreteçlerinin tanımlanmasında rastgelelik kavramının yanında en çok kullanılan terimlerden biri de entropidir. Bilgi entropisi bir veri kaynağının çıkışı ile ilgili

belirsizlik ölçüsünü ifade eder ve ilk defa 1949 yılında C. Shannon tarafından haberleşme teorisi alanında ortaya atılmıştır [10]. Sınıf sayısı L olan bir bilgi kaynağının, $X=(x_0,x_1,...,x_{L-1})$ entropisi Denklem (5.16)' da verilen ifade ile bulunur.

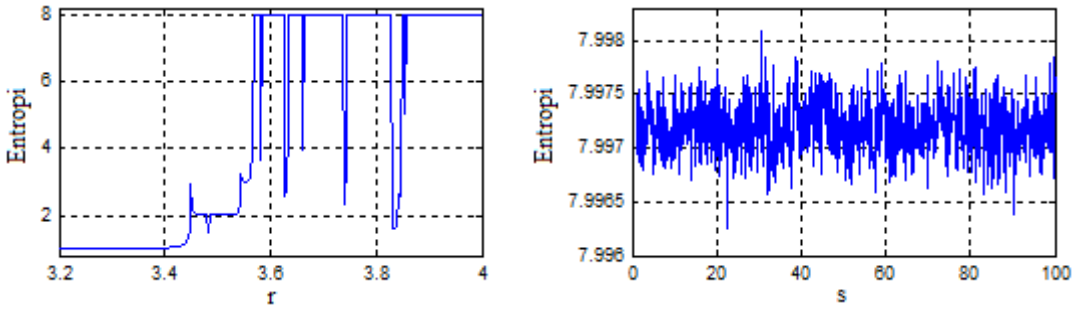
$$H(X)=-\sum_{i=0}^{L-1} p(x_i).\log_2 p(x_i) \tag{5.16}$$

Burada $p(x_i)$, x_i sembolünün olasılığını gösterirken, $H(X)$ entropisi ise bit ile ifade edilir. Ayrıca $H(X)$ ile L arasında Denklem (5.17)' deki gibi bir ilişki vardır.

$$0 \le H(X) \le \log_2 L \tag{5.17}$$

Buna göre $H(X)=0$ olması bir belirsizliğin olmadığını; $H(X)=\log_2 L$ olması ise serideki her bir sınıfın eşit olasılıkta yani $1/L$ olduğunu ve maksimumu belirsizliğin oluştuğunu gösterir. X serisinin tam bir tekdüzelik dağılımına sahip olması, yalnızca her sınıfın eşit sayıda olduğu yani kısaca maksimum entropi durumunda mümkündür. Örneğin, sınırlı bir aralıkta tamamen rastgele seriler üreten bir kaynak varsayılsın. Bu üreteç, $S=\{s_0,s_1,...,s_{255}\}$ gibi olsun ve tam tekdüzelik dağılım için 2^8 eleman içersin. Bu durumda, Denklem (5.16)' ya göre kaynağın entropisi $H(X)=8$ olarak hesaplanır ve bu değer maksimum ideal sonuçtur. Genel olarak pratik bilgi kaynakları için entropi değeri bu ideal değerden daha küçüktür. Sonuç olarak, bir sistemde ne kadar büyük bir entropi değeri varsa, o kadar çok rastgelelik veya belirsizlik hali söz konusudur. Dolayısıyla iyi bir rastgele sayı üretecinden mümkün olduğunca büyük entropi değeri beklenir.

İLH üretecinden elde edilen sayıların rastgelelik ve belirsizlik özellikleri ile ilgili bir değerlendirme yapılabilmesi amacıyla farklı s değerleri için gerçekleştirilen benzetimler sonucunda oluşan dağılımların entropi değerleri karşılaştırmalı olarak Şekil 5.10' da verilmiştir.



(a) (b)

Şekil 5.10 Üretilen kodlar için entropi analizi; (a) SLH, (b) İLH

Şekil 5.10(a)' da kolayca görüldüğü gibi SLH, yalnızca kaos durumundayken üretilen sayıların dağılımları ile ilgili entropi sonuçları ideal değere yakındır. Bunun dışındaki davranışlarda entropi çok küçük değerlere sahiptir çünkü bu durumlarda sistem periyodik ya da sabit davranış sergilemekte ve sonuçta oluşan dağılımın belirsizlik seviyesi de düşmektedir. Diğer taraftan İLH sisteminde ise tüm s değerleri altında hesaplanan entropi, ideal olan 8 değerine çok yakındır ve ortalama olarak 7.9974 seviyesindedir. Bu analiz sonucu ile birlikte sistem belirsizliği ile ilgili önemli bir sonuç daha çıkmaktadır. Sistemin uzun dönemli davranışıyla serilerin belirsizlik durumunun çok arttığı ve bunun sonucunda da entropi değerlerinin yükseldiği gözlemlenmiştir. Bu durum, Tablo 5.10' da ki sonuçlardan da görülebilmektedir. Bu sonuçların elde edilmesinde farklı ölçülerdeki karasel görüntülerin şifrelenmesi için görüntülerdeki piksel sayısı kadar anahtar kodlarının üretilmesi sağlanmış ve hesaplanan entropi değerleri Tablo 5.10' da verilmiştir. Sonuçların hesaplanmasında sistem parametreleri $s = 99$ ve $x_0 = 0.123$ gibi seçilmiştir.

Tablo 5.10 Entropi analiz sonuçları

Görüntü ölçüsü	Piksel sayısı	Entropi
128×128	16,384	7.987856
256×256	65,536	7.996888
512×512	262,144	7.999256
1024×1024	1.048,576	7.999819
2048×2048	4.194,304	7.999955
4096×4096	16.777,216	7.999986

Yapılan grafiksel analiz sonucuna göre İLH sisteminin belirsizlik performansı SLH ile karşılaştırıldığında yeni sistem belirsizlik performansı açısından çok daha iyi bir noktada

olduğu söylenebilir. Sayısal analiz sonuçlarına göre yeni sistemden üretilen kodların sayısı arttıkça entropi değerleri de artmaktadır ve bulunan tüm değerler teorik sonuca çok yakın seviyelerdedir. Kripto sistemde kullanılan şifreleme kodlarının tekdüzelik, duyarlılık ve belirsizlik analizleri başarıyla tamamlanmıştır. Ayrıca, İLH sisteminin çıkışına ait başarılı FIPS ve NIST rastgelelik analiz sonuçları da dikkate alındığında, İLH sisteminin davranışının, üretilen şifreleme kodları üzerindeki etkisini koruduğunu ve benzer karakteristik özellikleri anahtar serilerine yansıttığı görülmüştür. Bu anlamda, İLH üretecinden elde edilen anahtar kodlarının kriptografik işlemlerde kullanılabileceği açıktır.

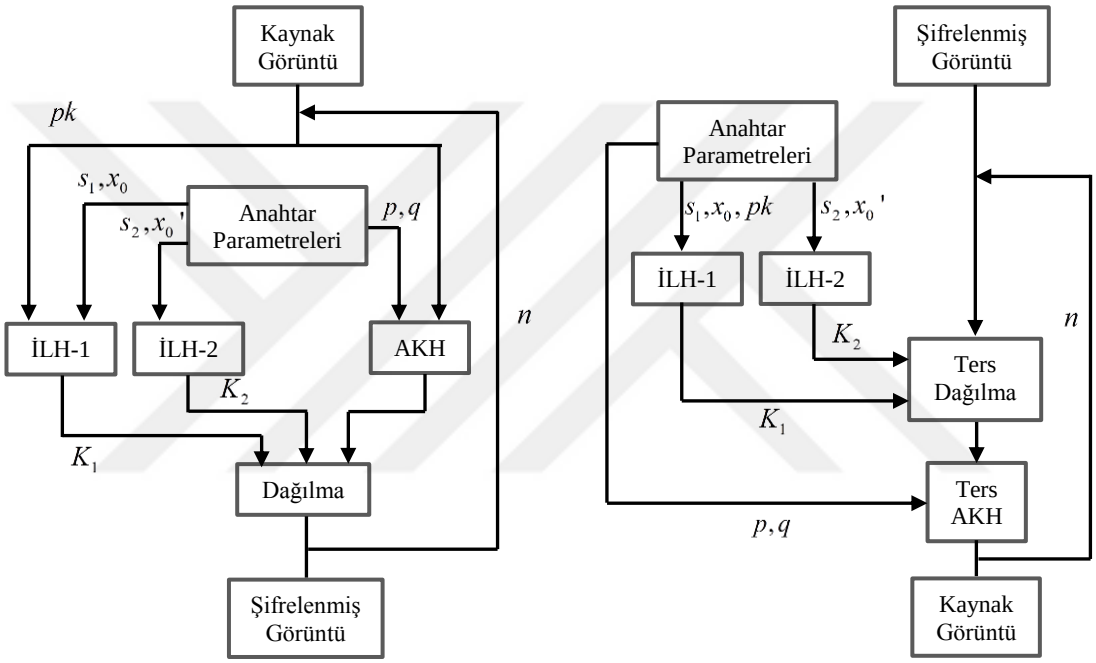
5.4. Sunulan Algoritmanın Çözme Kısmı

Kripto sistemde sunulan şifreleme yapısı, sırasıyla karıştırma ve dağılma kısımlarının istenilen sayıda tekrarlanmasıyla tamamlanmaktadır. Şifrelemede yapılan bu işlemlerin tam tersi, çözme algoritması olarak tanımlanır. Sunulan kripto sistem, simetrik şifreleme algoritmasına sahip olduğundan dolayı çözme işlemi için de aynı şifreleme anahtarlarına ihtiyaç vardır.

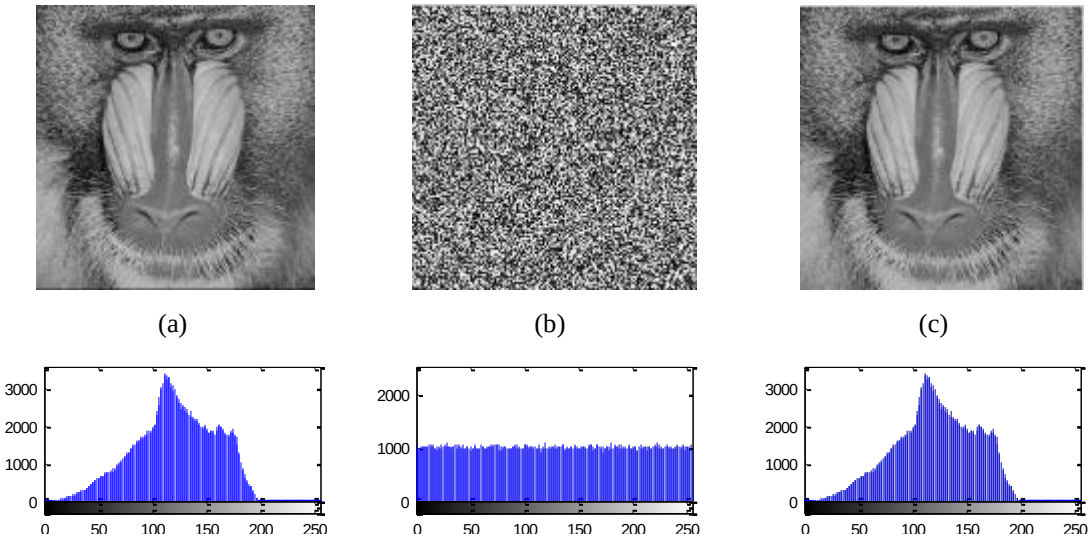
Şifreleme kodlarını üreten İLH-1 ve İLH-2 sistemleri kripto sistemin çözme kısmında da üreteç olarak kullanılmaktadır. Bu üreteçlerin parametreleri şifreleme anahtarı olarak düşünüldüğünden, aynı parametreler çözme tarafında da kullanıldığında tamamen aynı olan K_1 ve K_2 kodlarının üretilmesi mümkün olacaktır. Ancak burada şifreleme parametrelerinden $p, q, s_1, s_2, x_0\ x_0'$ ve n değerleri çözme algoritması içerisinde doğrudan kullanılabilirken, kaynak görüntüsü ile ilgili olan pk parametresi doğrudan kullanılamaz. Çünkü hesaplanmış bir pk parametresinin ters dönüşümü mümkün değildir. Daha açık bir ifadeyle, şifrelenmiş görüntüyü oluşturmada kullanılan pk parametresinin aynı şifrelenmiş görüntünün kullanılmasıyla hesaplanması olanaksızdır. Ayrıca, bu parametre değerinin bilinmeden şifreleme işleminde kullanılan anahtarların ve kodların elde edilemeyeceği de açıktır. Dolayısıyla pk değeri çözme işleminde ya özel olarak tanımlanmalı ya da pratik uygulamalarda ayrı bir güvenli kanal üzerinden çözme işleminin gerçekleşeceği sisteme iletilmelidir. Sunulan kripto sistemde bu durumla ilgili olarak birinci yaklaşım tercih edilmiştir. Kripto sistemde kullanılan şifreleme yapısının tam tersi görevi yapan çözme algoritması Denklem (5.18)' de verilmiştir.

$$p(i) = \left\{ (k_1(i) \oplus c(i)) + c(i-1) - k_2(i)^2 + 255^2 \right\} \bmod 256 \tag{5.18}$$

Kripto sistemde kullanılan şifreleme ve çözme algoritmalarına ait akış diyagramları Şekil 5.11' de gösterilmiştir. Şekil 5.11(a), sunulan kripto sistemin şifreleme kısmına ait akış diyagramını temsil ederken; Şekil 5.11(b) ise alıcı taraftaki çözme kısmını ifade eder. Test amaçlı 'Babun' görüntüsünün sunulan kripto sistemde kullanılmasıyla elde edilen şifrelenmiş görüntüsü Şekil 5.12' de gösterilmiştir. Burada şifreleme anahtarları: $s_1 = 21.864122$; $x_0 = 0.317548$; $s_2 = 46.392447$; $x_0' = 0.885642$; $p = 5$; $q = 4$ ve $n = 1$ olarak seçilmiştir.



Şekil 5.11 Sunulan kripto sisteminin akış şeması; (a) Şifreleme algoritması, (b) Çözme algoritması



Şekil 5.12 'Babun' görüntüsünün kripto sisteme uygulanışı; (a) Kaynak görüntü, (b) Şifrelenmiş görüntü, (c) Çözülen görüntü, (d) Kaynak görüntünün histogramı, (e) Şifrelenmiş görüntünün histogramı, (f) Çözülen görüntünün histogramı

6. KRİPTO SİSTEMİN PERFORMANS VE GÜVENLİK ANALİZİ

Bu kısımda, sunulan kripto sistemin güvenlik ve performans analizleri incelenecektir. İstatistiksel ve diferansiyel analizler kripto sistemin güvenlik değerlendirmesi ile ilgili yöntemleri içerir. Bu yöntemler kripto sistemle ilgili anahtar alanı analizi, anahtar hassasiyeti analizi, histogram analizi, entropi analizi, benzerlik analizi ve diferansiyel analiz tekniklerini kapsar. Ayrıca sunulan kripto sistemin, şifreleme ve çözme hızı, veri kaybı ve gürültü saldırısı altındaki tepkisi performans değerlendirilmesinde kullanılacak olan kriterlerdir.

Kripto sistemde oluşturulan şifrelenmiş görüntü üzerinden gerçekleştirilen diferansiyel analiz teknikleri, Ortalama Mutlak Hata (Mean Absolute Error, MAE), Piksel Sayıları Değişim Oranı (Number of Pixels Change Rate, NPCR), Birleşik Ortalama Değişim Yoğunluğu (Unified Average Changing Intensity, UACI), Tepe Sinyal-Gürültü Oranı (Peak Signal-to-Noise Ratio, PSNR), Şifreleme Kalitesi (Encryption Quality, EQ) ve Ortalama Karasel Hata (Mean Square Error, MSE) bu bölümde incelenmiştir.

6.1. Kripto Sistemin İstatistiksel Analizleri

6.1.1 Anahtar Alanı Analizi

Bir kripto sistemde kullanılabilecek farklı anahtarların tamamı kripto sistemin anahtar alanını oluşturur. Kripto sistemin anahtar alanı ölçüsü ise olabilecek toplam anahtarların sayısı kadardır. İdeal bir şifreleme algoritmasında anahtar alanı ölçüsünün 2^{100} değerinden büyük olması gerekir. Bu ölçü, aynı zamanda kaba-kuvvet saldırısının başarısız sayılmasında gerekli bir koşuldur.

Sunulan kripto sistemde karıştırma ve dağılma işlemleri sırayla uygulanmaktadır. Dolayısıyla kripto sistemin anahtar aralığı bu iki işlemin anahtar alanının çarpımıyla hesaplanır. Karıştırma işleminin anahtar alanı A_1; dağılma işleminin A_2 ve farklı iterasyonlarda aynı şifreleme anahtarları kullanılıyorsa, kripto sistemin toplam anahtar alanı Denklem (6.1)' deki gibi hesaplanır.

$$A = A_1 . A_2 \tag{6.1}$$

Eğer, her şifreleme iterasyonunda farklı anahtar kodları kullanılıyorsa o zaman kripto sistemin anahtar alanı, $A=(A_1.A_2)^n$ olur. Sunulan kripto sistemde ise karıştırma anahtarları sabit iken, dağılma kısmına ait anahtar kodları ise her bir iterasyonda değişmektedir. Dolayısıyla, kripto sisteminin toplam anahtar alanı, $A=A_1.A_2^n$ olacaktır. Kripto sistemde gerçekleştirilen şifreleme yapısı matematiksel olarak Denklem (6.2)' de gösterildiği gibi ifade edilebilir.

$$Y=\left[D(C(X,A_1),A_2)\right]^n \tag{6.2}$$

Burada X , Y sırasıyla kaynak ve şifrelenmiş görüntüyü; $C()$, $D()$ fonksiyonları sırasıyla karıştırma ve dağılma işlemlerini; n ise iterasyonu ifade eder. A_1 karıştırma anahtarlarını ve A_2 ise dağılma anahtarlarını gösterir. Ayrıca, $A_1=f\left(p,q\right)$ ve $A_2=f\left(s_1,s_2,x_0,x_0'\right)$ gibi değerlendirilebilir. Sunulan kripto sistemde, dağılma kısmına ait s_1,s_2,x_0 ve x_0' şifreleme anahtarları ondalıklı değerlerdir ve bunlar ikili sistemde 64-bit ile temsil edilebilmektedir. IEEE (Elektrik ve Elektronik Mühendisleri Enstitüsü) tarafından yayınlanan ondalıklı sayı standardına göre 64-bit bir sayı (double) için hesaplama tahmini yaklaşık olarak 10^{-15} dir. Dolayısıyla olası dağılma anahtarlarının toplam sayısı, $A_2=10^{60}$ olur.

Karıştırma kısmında ise kullanılan AKH için p,q parametrelerin, kaynak görüntünün ölçüsü olan N ile sınırlandırılması gerekir. Çünkü herhangi bir $a,b\in Z$ için $\left[1,(p+aN),(q+bN),(p+aN)(q+bN)+1\right]$ ile oluşturulmuş karışmış görüntü, $\left[1,p,q,pq+1\right]$ kullanarak oluşturulmuş görüntü ile tamamen aynıdır. Dolayısıyla p ve q, N sayıda muhtemel anahtar içerir. Sonuçta, karıştırma kısmı için anahtar alanı, $A_1=N^2$ olur. Her iki kısımda kullanılan anahtarlar birbirinden bağımsız olduğundan kripto sistemin iterasyon sayısından bağımsız olarak sahip olduğu toplam anahtar alanı, Denklem (6.3)' de verildiği gibi

$$A=A_1.A_2^n=N^2.10^{60}\approx 2^{200}.N^2 \tag{6.3}$$

olacaktır. Bu değer, kaba-kuvvet saldırısına karşı yeterli bir sonuçtur.

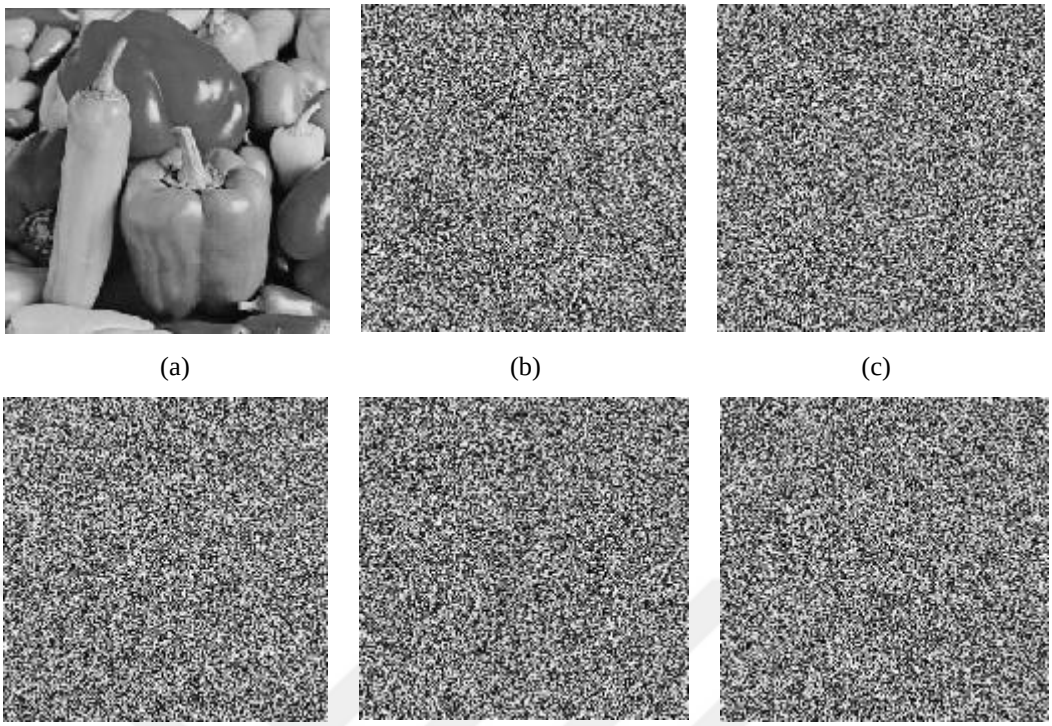
6.1.2 Anahtar Hassasiyeti Analizi

Güvenli kripto sistemler için yüksek anahtar hassasiyeti gereklidir ve iyi bir kripto sistem, kullanılan tüm şifreleme anahtarlarına karşı aşırı duyarlı olmalıdır. Bu anlamda, sunulan kripto sistem ele alındığında, AKH ve İLH sistemlerinin parametre hassasiyeti ne kadar yüksek olursa, kripto sistemin anahtar hassasiyeti de o kadar yüksek olur. Anahtar hassasiyeti analizi iki durum altında incelenir. Bunlardan birincisinde, birbirine çok yakın ama farklı şifreleme anahtarları, aynı kaynak görüntüyü şifrelemek için kullanıldığında tamamen farklı şifreli görüntülerin elde edilmesi beklenir. İkincisinde ise, çözme anahtarında çok küçük bir farkın oluşması durumunda şifrelenmiş görüntünün doğru şekilde çözümlenmemesi gerekir. Çok küçük anahtar farkından kaynaklı şifrelenmiş görüntünün değişme oranı matematiksel olarak ifade edilebilir. Buna göre, Denklem (6.4), karıştırma kısmında kullanılan şifreleme anahtarlarındaki küçük değişimin, şifrelenmiş görüntü üzerindeki değişme oranını ifade ederken; Denklem (6.5) ise dağılma kısmındaki değişimi vermektedir. Burada ΔA, anahtardaki küçük değişimi temsil eder.

$$\frac{\Delta Y}{Y} = \frac{\left| \left[D(C(X, A_1), A_2) \right]^n - \left[D(C(X, A_1 + \Delta A), A_2) \right]^n \right|}{Y} \tag{6.4}$$

$$\frac{\Delta Y}{Y} = \frac{\left| \left[D(C(X, A_1), A_2) \right]^n - \left[D(C(X, A_1), A_2 + \Delta A) \right]^n \right|}{Y} \tag{6.5}$$

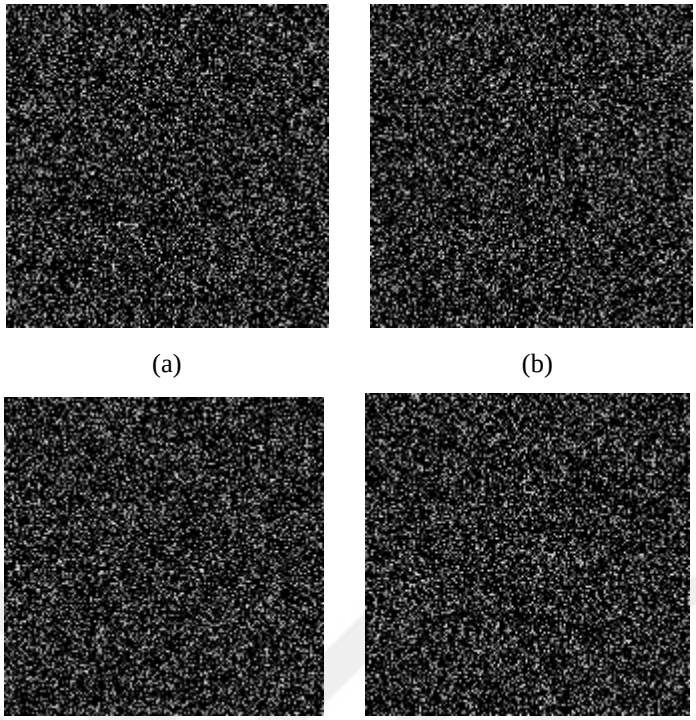
Birinci anahtar hassasiyeti analizi için, test görüntüsü olarak 256×256 ölçüsündeki 'Biber' görüntüsü kullanılmıştır. İlk olarak kaynak görüntü, rastgele seçilmiş şifreleme anahtarları: $s_1 = 85.246941872$; $x_0 = 0.417335966$; $s_2 = 37.881632551$; $x_0' = 0.794183248$; $p = 5$; $q = 4$ ve $n = 3$ kullanılarak kripto sistem tarafından şifrelenmiştir. Daha sonra dağılma anahtarlarına 10^{-9} gibi çok çok küçük bir değişim uygulanmış ve diğer parametre değerleri aynı kalarak şifreleme tekrarlanmıştır. Oluşan şifrelenmiş görüntüler ile şifrelenmiş bu görüntüler arasındaki fark görüntüleri sırasıyla Şekil 6.1 ve Şekil 6.2' de gösterilmiştir.



(a) (b) (c)

(d) (e) (f)

Şekil 6.1 Birinci duruma ait dağılma anahtarları hassasiyeti analizi; (a) 'Biber' test görüntüsü, (b) $s_1 = 85.246941872$; $x_0 = 0.417335966$; $s_2 = 37.881632551$; $x_0' = 0.794183248$; $p = 5$; $q = 4$ ve $n = 3$ kullanarak elde edilen şifrelenmiş görüntü, (c) $s_1 = 85.246941873$; $x_0 = 0.417335966$; $s_2 = 37.881632551$; $x_0' = 0.794183248$; $p = 5$; $q = 4$ ve $n = 3$ kullanarak elde edilen şifrelenmiş görüntü, (d) $s_1 = 85.246941872$; $x_0 = 0.417335967$; $s_2 = 37.881632551$; $x_0' = 0.794183248$; $p = 5$; $q = 4$ ve $n = 3$ kullanarak elde edilen şifrelenmiş görüntü, (e) $s_1 = 85.246941872$; $x_0 = 0.417335966$; $s_2 = 37.881632552$; $x_0' = 0.794183248$; $p = 5$; $q = 4$ ve $n = 3$ kullanarak elde edilen şifrelenmiş görüntü, (f) $s_1 = 85.246941872$; $x_0 = 0.417335966$; $s_2 = 37.881632551$; $x_0' = 0.794183249$; $p = 5$; $q = 4$ ve $n = 3$ kullanarak elde edilen şifrelenmiş görüntü



(a) (b)

(c) (d)

Şekil 6.2 Dağılma anahtarları hassasiyeti fark görüntüleri;
(a) Şekil 6.1(b) ve Şekil 6.1(c) arasındaki fark görüntüsü,
(b) Şekil 6.1(c) ve Şekil 6.1(d) arasındaki fark görüntüsü,
(c) Şekil 6.1(d) ve Şekil 6.1(e) arasındaki fark görüntüsü,
(d) Şekil 6.1(e) ve Şekil 6.1(f) arasındaki fark görüntüsü

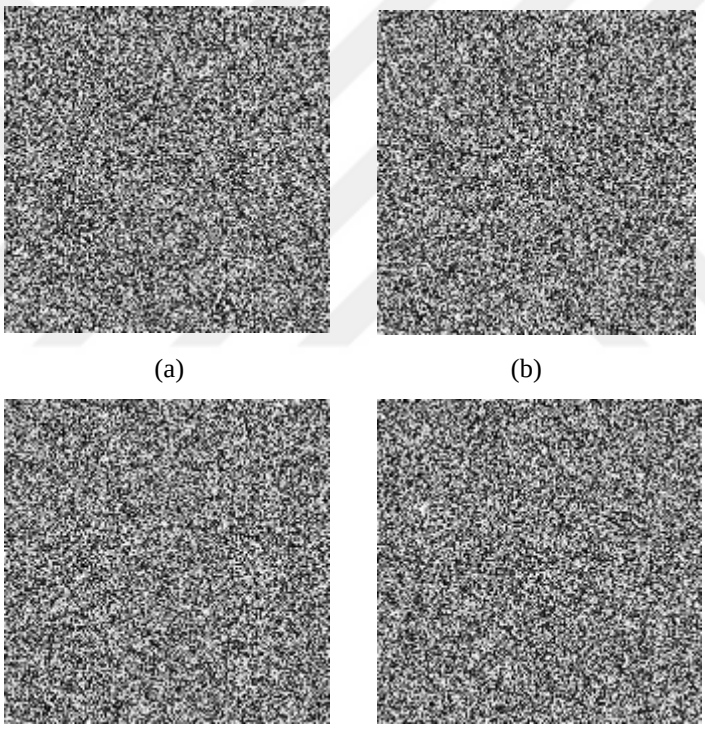
Şekil 6.2' deki görsel sonuçlardan anlaşılabileceği gibi dağılma anahtarlarının tamamında 10^{-9} gibi çok küçük bir değişimde bile tamamen farklı şifrelenmiş görüntüler elde edilmiştir. Ayrıca şifrelenmiş görüntüler için benzerlik katsayıları hesaplanmış ve sonuçlar Tablo 6.1' de verilmiştir. Bu sonuçlara göre şifrelenmiş görüntüler benzerlik yönünden birbirleriyle ilişkili değildir. Sunulan kripto sistemin dağılma anahtarlarına olan yüksek hassasiyetini, bu sonuçlar doğrulamaktadır.

Tablo 6.1 Şekil 6.1' deki şifrelenmiş görüntüler arasındaki benzerlik katsayıları

Şekil 6.1	**Benzerlik Katsayısı**
(b)-(c)	-0.000421
(c)-(d)	0.001898
(d)-(e)	0.006063
(e)-(f)	0.000212

Kripto sistem, dağılma anahtarları dışında aynı zamanda karıştırma anahtarlarındaki küçük bir değişime karşı da duyarlılık göstermesi gerekir. Bu anlamda, birinci anahtar hassasiyeti

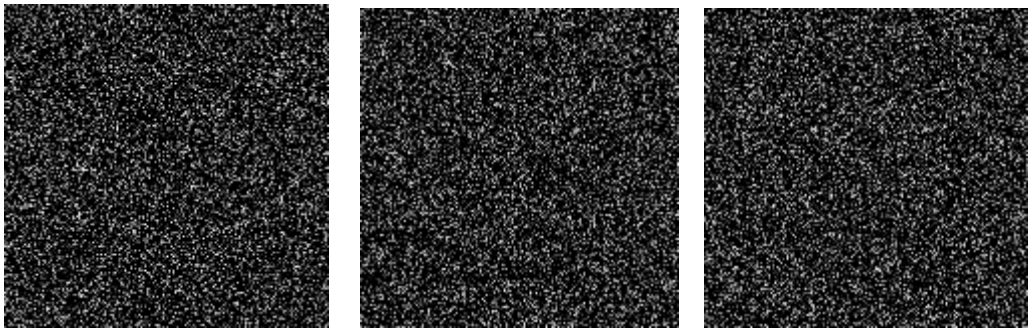
analizinde kullanılan dağılma anahtarları sabit kalarak karıştırma anahtarlarına küçük bir değişim uygulanmış ve aynı görüntü için şifreleme tekrarlanmıştır. Burada kullanılan şifreleme anahtarları ise Ş-1: $s_1 = 85.246941872$; $x_0 = 0.417335966$; $s_2 = 37.881632551$; $x_0' = 0.794183248$; $p = 5$; $q = 4$ ve $n = 3$; Ş-2: $s_1 = 85.246941872$; $x_0 = 0.417335966$; $s_2 = 37.881632551$; $x_0' = 0.794183248$; $p = 6$; $q = 4$ ve $n = 3$; Ş-3: $s_1 = 85.246941872$; $x_0 = 0.417335966$; $s_2 = 37.881632551$; $x_0' = 0.794183248$; $p = 5$; $q = 3$ ve $n = 3$; Ş-4: $s_1 = 85.246941872$; $x_0 = 0.417335966$; $s_2 = 37.881632551$; $x_0' = 0.794183248$; $p = 5$; $q = 4$ ve $n = 2$ gibidir. Şifreleme sonrasında oluşan şifrelenmiş görüntüler ile şifrelenmiş bu görüntüler arasındaki fark görüntüleri sırasıyla Şekil 6.3 ve Şekil 6.4' de verilmiştir.



(a) (b)

(c) (d)

Şekil 6.3 Birinci duruma ait karıştırma anahtarları hassasiyeti analizi; (a) $s_1 = 85.246941872$; $x_0 = 0.417335966$; $s_2 = 37.881632551$; $x_0' = 0.794183248$; $p = 5$; $q = 4$ ve $n = 3$ ile elde edilen şifrelenmiş görüntü, (b) $s_1 = 85.246941872$; $x_0 = 0.417335966$; $s_2 = 37.881632551$; $x_0' = 0.794183248$; $p = 6$; $q = 4$ ve $n = 3$ ile elde edilen şifrelenmiş görüntü, (c) $s_1 = 85.246941872$; $x_0 = 0.417335966$; $s_2 = 37.881632551$; $x_0' = 0.794183248$; $p = 5$; $q = 3$ ve $n = 3$ ile elde edilen şifrelenmiş görüntü, (d) $s_1 = 85.246941872$; $x_0 = 0.417335966$; $s_2 = 37.881632551$; $x_0' = 0.794183248$; $p = 5$; $q = 4$ ve $n = 2$ ile elde edilen şifrelenmiş görüntü



(a) (b) (c)

Şekil 6.4 Karıştırma anahtarları hassasiyeti fark görüntüleri; (a) Şekil 6.3(a) ve Şekil 6.3(b) arasındaki fark görüntüsü, (b) Şekil 6.3(b) ve Şekil 6.3(c) arasındaki fark görüntüsü, (c) Şekil 6.3(c) ve Şekil 6.3(d) arasındaki fark görüntüsü

Şekil 6.4' de verilen sonuçlara göre sunulan kripto sistem benzer şekilde karıştırma anahtarlarına da yüksek oranda duyarlılık göstermiştir. Birbirine çok yakın karıştırma anahtarlarıyla elde edilen şifrelenmiş görüntüler arasındaki benzerlik katsayıları Tablo 6.2' de verilmiştir. Bu sonuçlar, şifrelenmiş görüntülerin birbirinden tamamen farklı olduğunu ifade etmektedir.

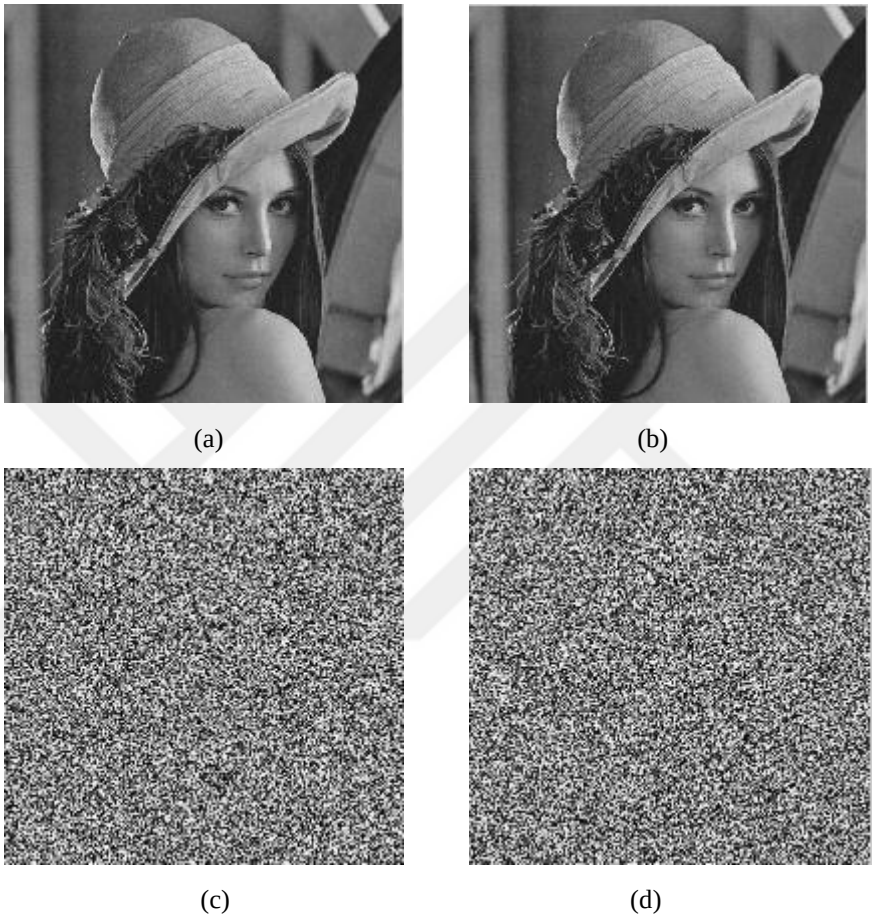
Tablo 6.2 Şekil 6.3' deki şifrelenmiş görüntüler arasındaki benzerlik katsayıları

Şekil 6.3	**Benzerlik Katsayısı**
(a)-(b)	0.001709
(b)-(c)	0.004726
(c)-(d)	-0.000592

Sonuç olarak sunulan kripto sistem, tüm şifreleme anahtarlarına etkin bir şekilde bağlılık gösterdiğinden birinci durumdaki anahtar hassasiyeti analizini başarıyla tamamlamıştır. Ayrıca bu sonuçlar, kripto sistemin diferansiyel saldırılara karşı dayanıklılığını da göstermektedir.

Kripto sistemde gerçekleştirilen şifreleme algoritması daha önceden de belirtildiği üzere kullanılan kaynak görüntüye karşı hassas duyarlıdır. Bu duyarlılığın şifrelenmiş görüntü üzerindeki etkisini incelemek için aynı ölçülere (512×512) sahip iki 'Lena' test görüntüsü kripto sistemde kullanılmıştır. Önce 'Lena' görüntülerinden biri olan P_1, $s_1 = 85.246941872$; $x_0 = 0.417335966$; $s_2 = 37.881632551$; $x_0' = 0.794183248$; $p = 5$; $q = 4$ ve $n = 3$ anahtarı ile şifrelenerek C_1 görüntüsü elde edilmiştir. Daha sonra diğer 'Lena' görüntüsünün (P_2) tam orta noktasındaki piksel, $P(256,256) = 89$ değerinde bir bit değişim

yapılarak yeni değer $P(256,256)=90$ yapılmış ve aynı şifreleme anahtarı ile yeni görüntü şifrelenerek C_2 görüntüsü elde edilmiştir. C_1 ve C_2 ile temsil edilen şifrelenmiş görüntüler Şekil 6.5' de gösterilirken, bu görüntüler arasındaki benzerlik katsayılarıyla ilgili sonuçlar ise Tablo 6.3' de verilmiştir.



(a) (b)

(c) (d)

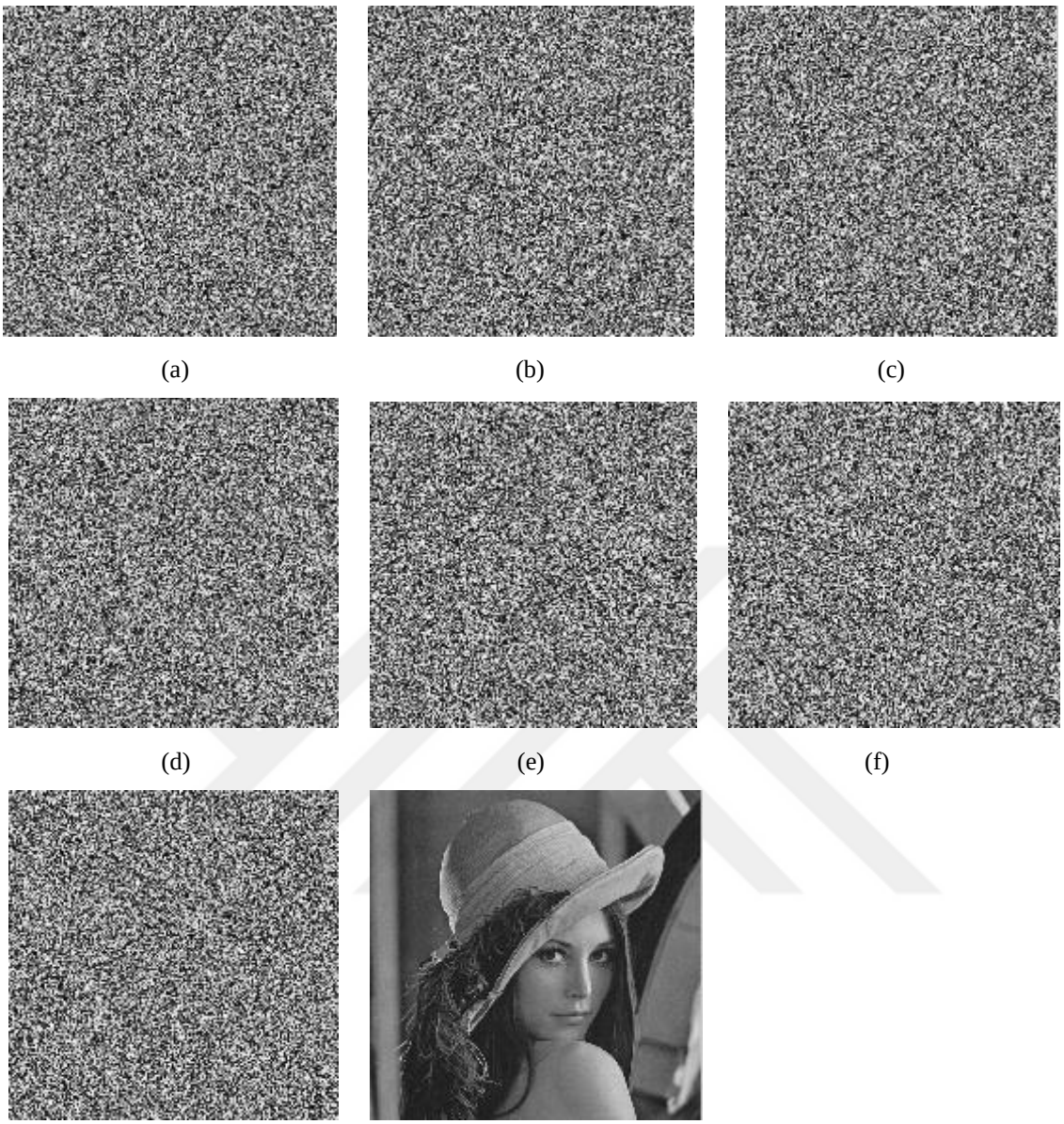
Şekil 6.5 Kaynak görüntüye bağlılık analizi; (a) Orijinal 'Lena' görüntüsü (P_1), (b) Orta piksel değeri bir bit değiştirilmiş 'Lena' görüntüsü (P_2), (c) P_1 görüntüsüne karşılık gelen şifrelenmiş C_1 görüntüsü, (d) P_2 görüntüsüne karşılık gelen şifrelenmiş C_2 görüntüsü

Tablo 6.3 Şifrelenmiş ‘Lena’ görüntüleri arasındaki benzerlik katsayıları ve *pk* değerleri

Şekil 6.5	***pk* değeri**	**Benzerlik Katsayısı**
(a)	22.982421	-
(b)	22.978515	-
(c)	274.837890	-
(d)	102.728515	-
(a)-(b)	-	0.999999
(c)-(d)	-	-0.004822
(a)-(c)	-	0.000793
(b)-(d)	-	-0.002622

Kripto sistemde aynı şifreleme anahtarları kullanarak kaynak görüntünün herhangi bir pikselinde bir bit değişimde bile tamamen farklı şifreli görüntülerin elde edildiği durum, Tablo 6.3’ de verilen sonuçlardan anlaşılmaktadır. Bu sonuçlar, sunulan kripto sistemin şifreleme anahtarları dışında kaynak görüntüye de hassas duyarlılık sergilediğini göstermektedir.

İkinci anahtar hassasiyeti analizi için kripto sistemin çözme kısmında kullanılan anahtara çok küçük bir değişim uygulandığında şifrelenmiş görüntünün doğru bir şekilde çözümlenmemesi gerekir. Bu analiz için Şekil 6.5(c)’ de verilen şifrelenmiş ‘Lena’ görüntüsü kullanılmıştır. Dolayısıyla çözme işlemi için sırasıyla, Ç-1: $s_1 = 85.246941871$; $x_0 = 0.417335966$; $s_2 = 37.881632551$; $x_0' = 0.794183248$; $p = 5$; $q = 4$ ve $n = 3$; Ç-2: $s_1 = 85.246941872$; $x_0 = 0.417335965$; $s_2 = 37.881632551$; $x_0' = 0.794183248$; $p = 5$; $q = 4$ ve $n = 3$; Ç-3: $s_1 = 85.246941872$; $x_0 = 0.417335966$; $s_2 = 37.881632550$; $x_0' = 0.794183248$; $p = 5$; $q = 4$ ve $n = 3$; Ç-4: $s_1 = 85.246941872$; $x_0 = 0.417335966$; $s_2 = 37.881632551$; $x_0' = 0.794183247$; $p = 5$; $q = 4$ ve $n = 3$; Ç-5: $s_1 = 85.246941872$; $x_0 = 0.417335966$; $s_2 = 37.881632551$; $x_0' = 0.794183248$; $p = 4$; $q = 4$ ve $n = 3$; Ç-6: $s_1 = 85.246941872$; $x_0 = 0.417335966$; $s_2 = 37.881632551$; $x_0' = 0.794183248$; $p = 5$; $q = 3$ ve $n = 3$; Ç-7: $s_1 = 85.246941872$; $x_0 = 0.417335966$; $s_2 = 37.881632551$; $x_0' = 0.794183248$; $p = 5$; $q = 4$ ve $n = 2$ anahtarları kullanılmıştır. Çözme işlemine ait görsel sonuçlar Şekil 6.6’ da verilmiştir.



(a) (b) (c)

(d) (e) (f)

(g) (h)

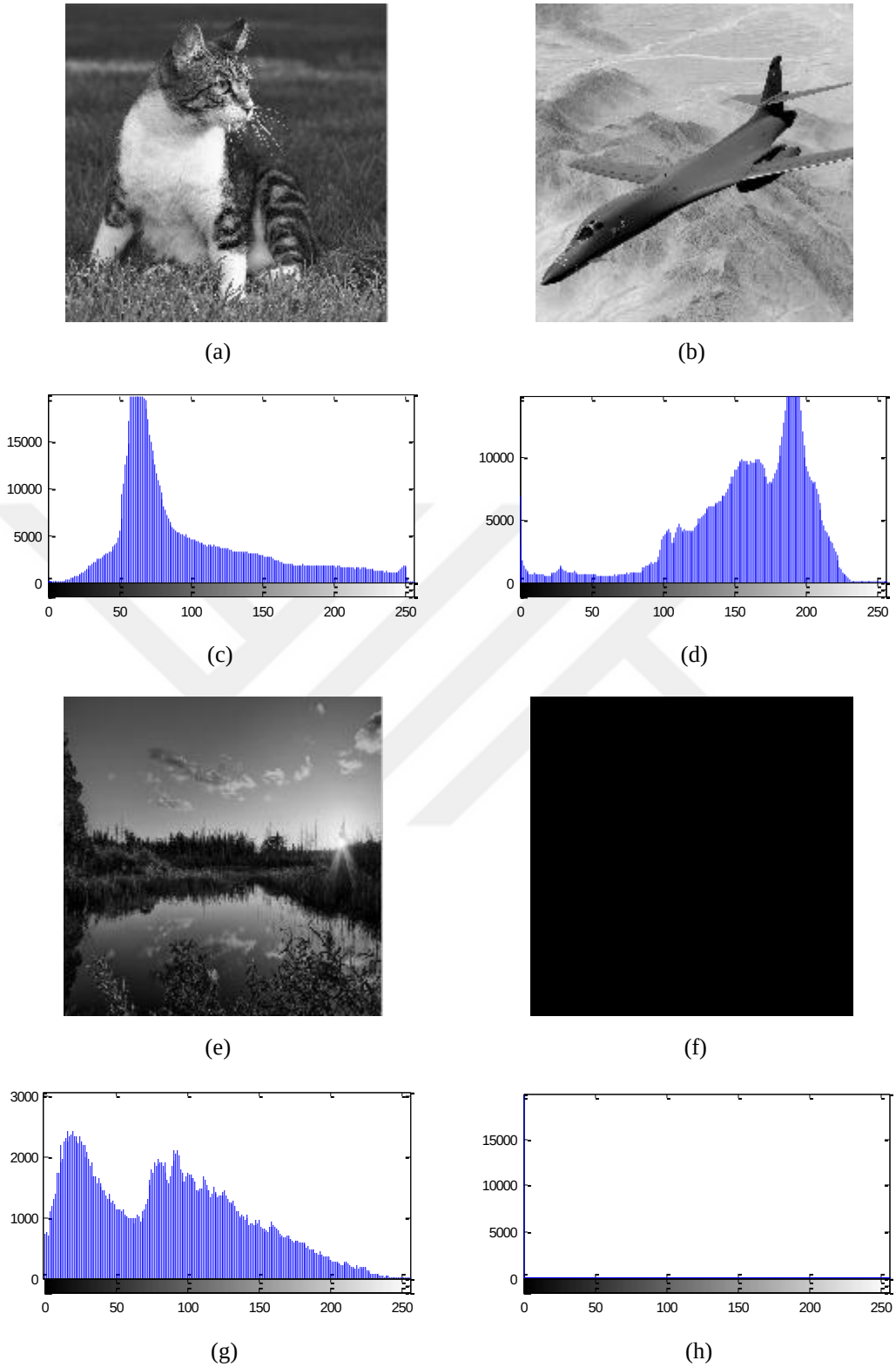
Şekil 6.6 İkinci duruma ait şifreleme anahtarları hassasiyeti analizi; (a) Ç-1 anahtarı kullanarak yanlış çözülmüş görüntü, (b) Ç-2 anahtarı kullanarak yanlış çözülmüş görüntü, (c) Ç-3 anahtarı kullanarak yanlış çözülmüş görüntü, (d) Ç-4 anahtarı kullanarak yanlış çözülmüş görüntü, (e) Ç-5 anahtarı kullanarak yanlış çözülmüş görüntü, (f) Ç-6 anahtarı kullanarak yanlış çözülmüş görüntü, (g) Ç-7 anahtarı kullanarak yanlış çözülmüş görüntü, (h) $s_1 = 85.246941872$; $x_0 = 0.417335966$; $s_2 = 37.881632551$; $x_0' = 0.794183248$; $p = 5$; $q = 4$ ve $n = 3$ anahtarı kullanarak doğru çözülmüş görüntü

Benzer şekilde kripto sistemin çözme anahtarında da 10^{-9} gibi çok küçük bir hata olması durumunda, şifrelenmiş görüntünün doğru bir şekilde çözülemeyeceği, Şekil 6.6' da verilen sonuçlardan kolayca anlaşılabilmektedir. Sonuç olarak sunulan kripto sistem şifreleme ve çözme işlemlerinde kullanılan anahtarlara karşı çok duyarlı olup, bunlara yüksek hassasiyet göstermektedir.

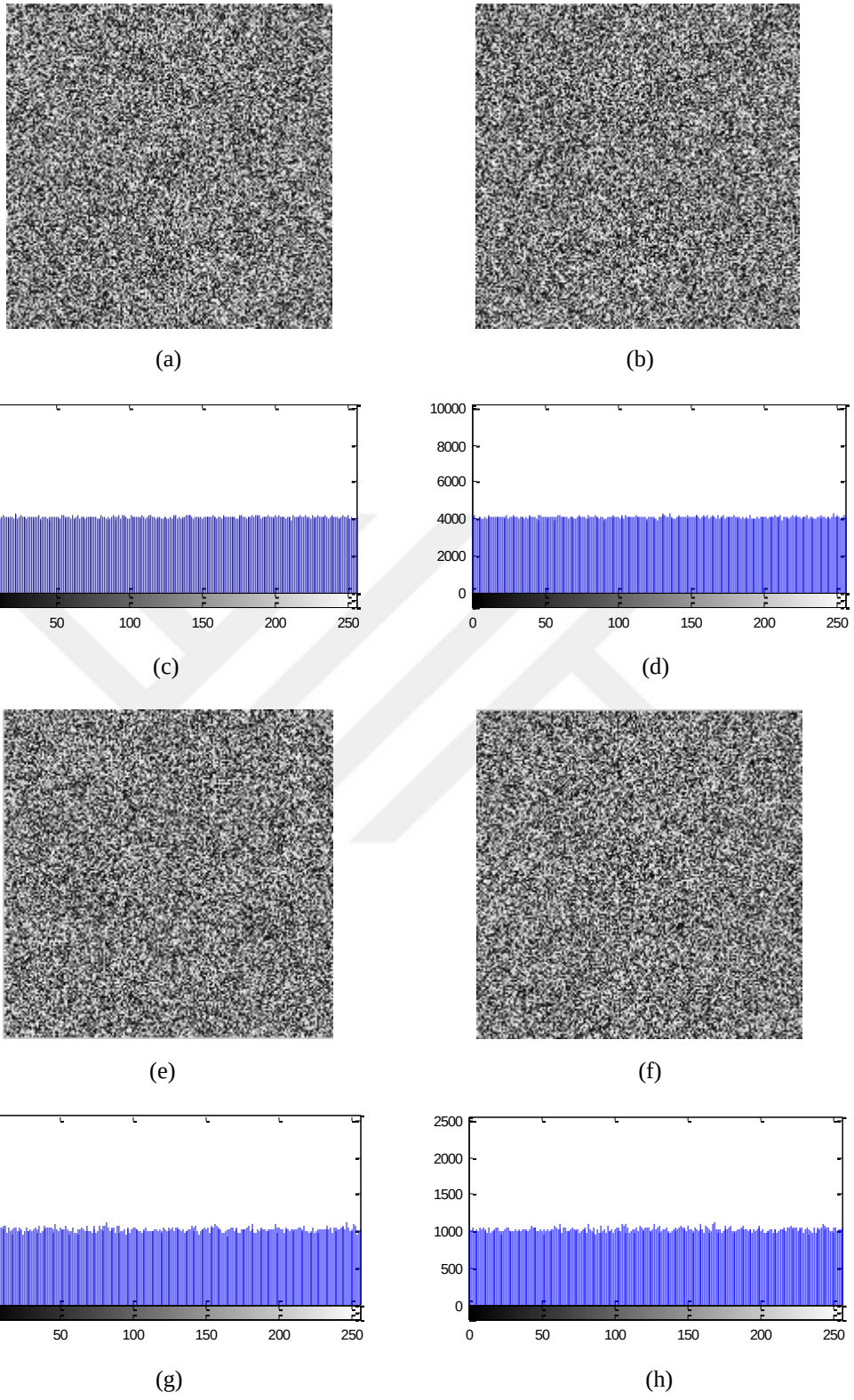
6.1.3 Histogram Analizi

Bir görüntü içerisindeki piksel değerlerinin dağılımını temsil etmede histogram grafiği kullanılır. Şifrelenmiş görüntü içerisindeki bir piksel değerin eşit olasılıkla dağılımı tekdüze bir histogram oluşturur ve böyle bir dağılım güvenlik açısından istatistiksel saldırılara karşı daha dayanıklıdır. Dolayısıyla, şifrelenmiş bir görüntüye ait ideal histogram grafiği yeterince tekdüze bir yapıda ve kaynak görüntüsündekinden tamamen farklı bir formda olması gerekir.

Histogram analizi için farklı ölçülere sahip yirmiden fazla test görüntüsü kullanılmıştır. Elde edilen tüm grafiksel sonuçlar, burada verilen sonuçlar ile benzerlik göstermektedir. Sınırlı yer alanından dolayı test görüntülerin tamamına ait histogram sonuçları burada paylaşılmamıştır. Burada, piksel dağılımları birbirinden çok farklı olan dört test görüntüsü (Kedi, Uçak, Göl ve Siyah) ele alınmıştır. Bu görüntüler ilk olarak sunulan kripto sistemde şifrelenmiş ve daha sonra şifreli görüntülerle ilgili histogram grafikleri elde edilmiştir. Kaynak ve şifrelenmiş görüntülere ait histogram sonuçları sırasıyla Şekil 6.7 ve Şekil 6.8’ de verilmiştir.



Şekil 6.7 Farklı test görüntülerin histogram grafikleri; (a) 'Kedi' görüntüsü (1024×1024), (b) 'Uçak' görüntüsü (1024×1024), (c) 'Kedi' görüntüsünün histogramı, (d) 'Uçak' görüntüsünün histogramı, (e) 'Göl' görüntüsü (512×512), (f) 'Siyah' görüntüsü (512×512), (g) 'Göl' görüntüsünün histogramı, (h) 'Siyah' görüntüsünün histogramı



Şekil 6.8 Şifrelenmiş görüntülerin histogram analizleri; (a) Şifrelenmiş 'Kedi' görüntüsü, (b) Şifrelenmiş 'Uçak' görüntüsü, (c) Şifrelenmiş 'Kedi' görüntüsünün histogramı, (d) Şifrelenmiş 'Uçak' görüntüsünün histogramı, (e) Şifrelenmiş 'Göl' görüntüsü, (f) Şifrelenmiş 'Siyah' görüntüsü, (g) Şifrelenmiş 'Göl' görüntüsünün histogramı, (h) Şifrelenmiş 'Siyah' görüntüsünün histogramı

Şekil 6.8’ de verilen şifrelenmiş görüntülere ait histogram sonuçları, kaynak görüntülerin histogram grafiklerinden tamamen farklıdır. Test görüntüsü olarak tamamen tek renk piksel değerinden oluşan ‘Siyah’ görüntüsü bile kripto sistemde kullanıldığında, oluşan şifrelenmiş görüntüdeki piksel dağılımları tekdüze bir yapıda oluşmuş ve piksel değerleri hemen hemen eşit olasılıkla dağılmıştır. Gerçekleştirilen diğer histogram analizlerinde farklı şifreleme anahtarları ve iterasyon sayıları kullanılmıştır. Bu analizlerde de, şifrelenmiş görüntülere ait düzenli bir histogram dağılımı elde edilmiştir. Bu anlamda, sunulan kripto sistem istatistiksel saldırılara karşı dayanıklıdır.

6.1.4 Entropi Analizi

Genel olarak bir sistemde belirsizlik veya rastgelelik fazlaysa sistem o kadar fazla entropi içerir. Bir bilgi kaynağının entropisi, Denklem (5.16)’ da verildiği gibi hesaplanır. Burada, entropi değeri ne kadar yüksek ise bu kaynak ile ilgili belirsizlik ve öngörülemezlik o kadar yüksek olur. Şifrelenmiş görüntüler de kaynak görüntüler gibi 8-bit olduğundan dolayı şifrelenmiş bir görüntüye ait maksimum entropi değeri 8 olabilir.

Bu kısımda, şifrelenmiş görüntülerin belirsizliği ve rastgeleliği entropi analiziyle incelenmiştir. Bunun için 10 farklı test görüntüsü kullanılarak elde edilmiş şifrelenmiş görüntülerin entropi değerleri hesaplanmıştır. Burada her bir görüntü için rastgele seçilmiş farklı şifreleme anahtarları kullanılmıştır. Şifrelenmiş görüntülere ait farklı iterasyon değerlerindeki entropi sonuçları Tablo 6.4’ de verilmiştir.

Tablo 6.4 Şifrelenmiş görüntülerin entropi sonuçları

Ölçü	**Test görüntüsü**	**Kaynak Entropisi**	**Şifrelenmiş görüntü Entropisi**				
			n=1	n=2	n=3	n=4	n=5
2048×2048	Kaplan	7.7066625	7.9999545	7.9999537	7.9999523	7.9999490	7.9999514
1024×1024	Dünya	6.2623288	7.9998089	7.9998224	7.9998366	7.9998343	7.9998281
1024×1024	Park	7.2862715	7.9998276	7.9998146	7.9998399	7.9998283	7.9998033
1024×1024	Kedi	7.2068275	7.9998192	7.9998269	7.9998058	7.9998495	7.9998503
512×512	Kurbağa	6.9941440	7.9992348	7.9993463	7.9992591	7.9993381	7.9994659
512×512	Manzara	7.3941130	7.9991954	7.9993160	7.9994143	7.9993356	7.9993582
512×512	Lena	7.2297834	7.9993070	7.9993062	7.9992419	7.9992959	7.9992326
256×256	Babun	7.2200285	7.9972853	7.9970487	7.9971558	7.9974587	7.9970716
256×256	Ay	5.3544761	7.9974566	7.9974513	7.9974127	7.9973389	7.9970833
256×256	Biber	7.5746110	7.9973912	7.9974311	7.9969896	7.9974457	7.9971429
		Ortalama	7.9989281	7.9989317	7.9988908	7.9989674	7.9988788

Şifrelenmiş görüntüler üzerinde gerçekleştirilen Entropi analizlerinde maksimum değer olarak 7.9999896 seviyesine ulaşılmıştır. Tablo 6.4' de verilen entropi sonuçlarının ideal değere çok yakın olması, kripto sistemin rastgele ve belirsiz bir çıkış üretmesiyle yorumlanabilir. Bu anlamda, sunulan şifreleme algoritması entropi saldırılarına karşı güvenlidir.

6.1.5 Benzerlik Analizi

Anlamlı bir görüntüde birbirine yakın komşu pikseller arasında benzerlik yönünden güçlü bir ilişki vardır. Bu durum, yakın koordinattaki piksellerin değerce birbirine yakın olmasıyla da açıklanabilir. Dolayısıyla iyi şifrelenmiş bir görüntüde yakın piksel değerleri birbirinden çok farklı olur. Kripto sistemde farklı test görüntüleri kullanarak elde edilmiş şifrelenmiş görüntülerin yatay, dikey ve köşegen yönlerindeki komşu pikseller arasındaki benzerlik katsayıları sırasıyla Tablo 6.5-6.7' de gösterilmiştir. On farklı test görüntüsü kullanarak elde edilen bu sonuçlar, şifreleme öncesindeki komşu pikseller arasındaki yakın benzerliğin, şifreleme sonrasında tamamen kırıldığını göstermektedir.

Şifreleme sonrasında oluşan şifrelenmiş görüntü ile kaynak görüntü arasında da bir benzerliğin bulunmaması gerekir. Bu anlamda, şifrelenmiş görüntü ile kaynak görüntü arasındaki benzerlik katsayı değerinin sıfıra yakın olması, bu iki görüntünün birbirinden çok farklı olması şeklinde yorumlanır. Kaynak ve şifrelenmiş görüntüler arasındaki benzerlik analizi için farklı ölçülere sahip test görüntüleri kullanılmıştır. Bu analizde, farklı iterasyon sayılarındaki benzerlik katsayıları hesaplanmış ve sonuçlar Tablo 6.8' de verilmiştir.

Tablo 6.5 Yatay yöndeki komşu pikseller arasındaki benzerlik katsayıları

Ölçü	Test görüntüsü	Kaynak Görüntü	Şifrelenmiş Görüntü				
			n=1	n=2	n=3	n=4	n=5
2048×2048	Kaplan	0.97699	-0.012703	0.024028	-0.016310	0.001804	-0.008156
1024×1024	Dünya	0.83944	0.004463	-0.042323	-0.002347	0.028593	0.015531
1024×1024	Kedi	0.99750	-0.015029	0.035351	0.020723	-0.025320	0.030782
1024×1024	Manzara	0.96716	0.038868	-0.024129	-0.026468	0.021957	0.042578
512×512	Göl	0.92656	0.014210	0.017099	0.006269	-0.008895	0.048696
512×512	Lena	0.94207	-0.075737	-0.039164	0.012965	-0.045137	0.002043
512×512	Özgürlük Anıtı	0.97628	0.065107	-0.019050	-0.014135	0.038311	0.005956
256×256	Kameraman	0.93457	-0.079034	-0.013721	0.008349	-0.114143	0.027769
256×256	Babun	0.94441	-0.068765	-0.042538	-0.068416	-0.003879	-0.017103
256×256	Biber	0.95186	-0.019327	-0.024301	0.015006	0.086278	0.054997

Tablo 6.6 Dikey yöndeki komşu pikseller arasındaki benzerlik katsayıları

Ölçü	Test görüntüsü	Kaynak Görüntü	Şifrelenmiş Görüntü				
			n=1	n=2	n=3	n=4	n=5
2048×2048	Kaplan	0.99768	-0.018915	0.010080	0.011588	-0.026223	0.010974
1024×1024	Dünya	0.97244	0.039977	-0.050183	-0.031257	0.036236	0.039447
1024×1024	Kedi	0.99317	0.005250	-0.021753	-0.041863	0.021080	-0.071605
1024×1024	Manzara	0.97651	-0.031033	0.024227	-0.022308	-0.004635	-0.026325
512×512	Göl	0.98918	0.005357	-0.055570	-0.049019	-0.021805	0.036580
512×512	Lena	0.96042	0.087150	0.008043	-0.084234	0.027557	-0.002003
512×512	Özgürlük Anıtı	0.98186	0.026691	-0.077255	0.006536	-0.015064	-0.022385
256×256	Kameraman	0.89732	-0.026605	0.051436	-0.094164	0.022925	0.042216
256×256	Babun	0.89967	0.003695	-0.012967	-0.085827	-0.002488	-0.003450
256×256	Biber	0.88688	-0.031779	-0.034018	-0.092287	0.048079	-0.044158

Tablo 6.7 Köşegen yöndeki komşu pikseller arasındaki benzerlik katsayıları

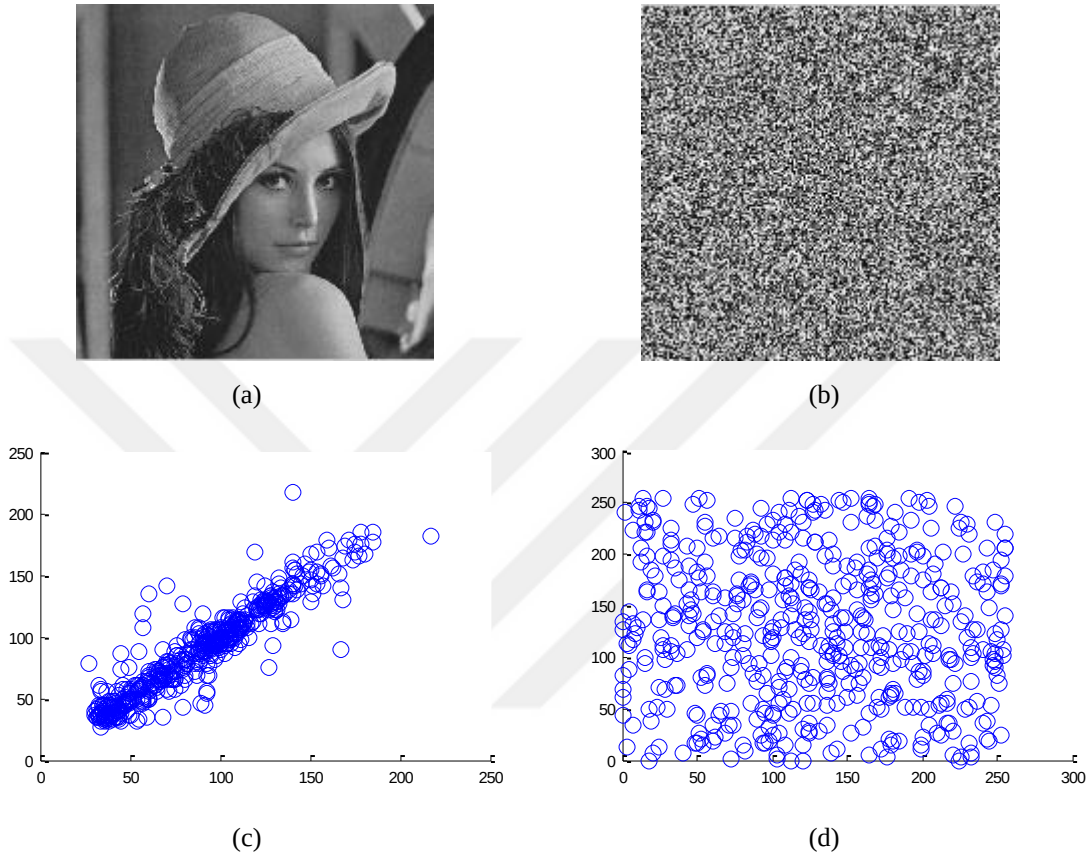
Ölçü	Test görüntüsü	Kaynak Görüntü	Şifrelenmiş Görüntü				
			n=1	n=2	n=3	n=4	n=5
2048×2048	Kaplan	0.98231	-0.035408	-0.009294	0.001575	-0.005490	-0.012544
1024×1024	Dünya	0.88721	0.010248	-0.008210	0.021665	0.003590	-0.045521
1024×1024	Kedi	0.96585	-0.023899	0.008807	-0.021373	0.013213	-0.036364
1024×1024	Manzara	0.97320	0.042289	0.014680	-0.015575	-0.000983	-0.004105
512×512	Göl	0.91707	0.005327	-0.049632	-0.035401	0.064142	0.058523
512×512	Lena	0.93520	0.053649	0.043698	0.009740	-0.019768	0.003751
512×512	Özgürlük Anıtı	0.95275	-0.021418	-0.045580	-0.048146	-0.079396	0.013585
256×256	Kameraman	0.89237	0.023987	-0.013109	0.108136	-0.004487	0.006154
256×256	Babun	0.80291	0.053246	0.063464	-0.015715	-0.083619	0.042647
256×256	Biber	0.92517	-0.052804	0.032521	-0.005192	0.081685	-0.077951

Tablo 6.8 Kaynak görüntüler ile şifrelenmiş görüntüler arasındaki benzerlik katsayıları

Ölçü	Test görüntüsü	Benzerlik katsayısı				
		n=1	n=2	n=3	n=4	n=5
2048×2048	Kaplan	0.000603	-0.000009	-0.000583	-0.000408	-0.000109
1024×1024	Dünya	0.002769	0.001194	-0.000058	0.000546	0.001277
1024×1024	Kedi	0.000186	0.000183	-0.001335	0.000120	-0.000814
1024×1024	Manzara	-0.000933	0.000246	-0.000741	-0.000464	-0.001727
512×512	Göl	0.002433	0.000632	-0.004292	-0.000243	-0.002618
512×512	Lena	0.002466	-0.003388	0.001675	-0.004237	-0.001390
512×512	Özgürlük Anıtı	-0.000757	-0.002153	0.000516	-0.001660	0.003603
256×256	Kameraman	-0.003202	-0.001856	-0.002859	0.000628	-0.002453
256×256	Babun	0.005409	0.003139	0.000234	0.005215	-0.007618
256×256	Biber	-0.003312	0.001951	-0.004140	0.001998	0.000031

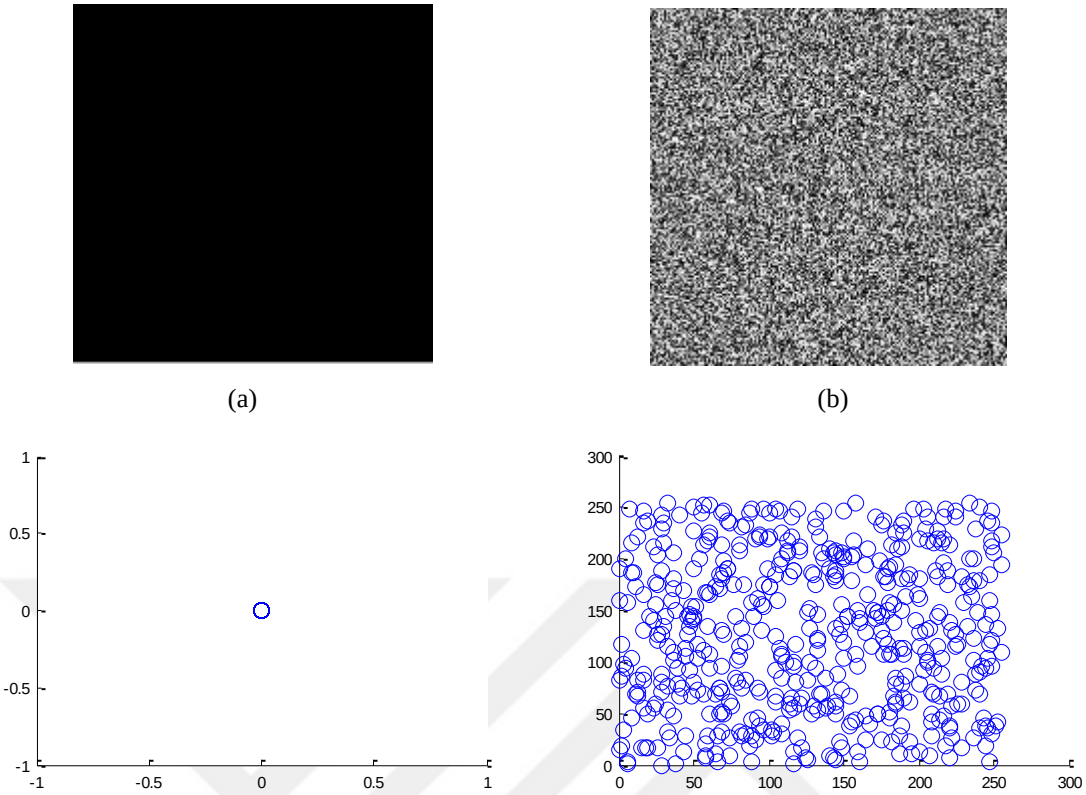
Benzerlik analizi sonuçlarına göre şifrelenmiş görüntüler içerisindeki tüm yöndeki yakın pikseller arasında benzerlik yönünden bir yakınlığın olmadığı ayrıca şifrelenmiş görüntüler ile kaynak görüntüler arasındaki benzerlik katsayı değerlerinin sıfıra çok yakın olduğu

kolayca görülebilmektedir. Şekil 6.9 ve Şekil 6.10 sırasıyla ‘Lena’ ve ‘Siyah’ test görüntülerin şifrelenmeden önceki ve sonraki durumlarına ait köşegen yönündeki piksellerin ilişki dağılımlarını göstermektedir.



Şekil 6.9 (a) ‘Lena’ görüntüsü, (b) Şifrelenmiş ‘Lena’ görüntüsü, (c) ‘Lena’ görüntüsüne ait köşegen yönündeki piksellerin ilişki dağılımı, (d) Şifrelenmiş ‘Lena’ görüntüsüne ait köşegen yönündeki piksellerin ilişki dağılımı

Şekil 6.9(c)’ de verilen ‘Lena’ görüntüsünde dikkat edilirse pikseller arasında doğrusal bir dağılım gerçekleşirken, şifrelenmiş görüntüsünde ise rastgele bir dağılım oluşmuş ve bu dağılım tüm alanı kaplamıştır. Benzer durum Şekil 6.10’ da gösterilen ve tüm piksel değerleri 0 olan ‘Siyah’ görüntüsü için de geçerlidir. Bu sonuç, şifreleme sonrası herhangi bir piksel değeri ile yakınındaki komşu piksel değerinin çok düşük bir ilişkiye sahip olduğunu gösterir.



Şekil 6.10 (a) 'Siyah' görüntüsü, (b) Şifrelenmiş 'Siyah' görüntüsü, (c) 'Siyah' görüntüsüne ait köşegen yönündeki piksellerin ilişki dağılımı, (d) Şifrelenmiş 'Siyah' görüntüsüne ait köşegen yönündeki piksellerin ilişki dağılımı

6.2. Kripto Sistemin Diferansiyel Analizleri

6.2.1 Ortalama Mutlak Hata Analizi

Ortalama Mutlak Hata (MAE) testi, kaynak görüntüdeki değişimin şifrelenmiş görüntü üzerindeki hassasiyetini değerlendiren bir yöntemdir. MAE değeri şifrelenmiş görüntünün, kaynak görüntüye göre ne kadar değiştiğini belirtir ve Denklem (6.6)' daki gibi tanımlanır.

$$MAE = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} |C(i, j) - P(i, j)| \tag{6.6}$$

Burada, (M, N) görüntü ölçüsünü gösterirken; $C(i, j)$ ve $P(i, j)$ sırasıyla şifrelenmiş ve kaynak görüntünün (i, j) pozisyonundaki piksel değerini temsil eder. MAE değerinin

büyük olması, şifreleme güvenliği açısından önemlidir. Dolayısıyla daha büyük MAE değeri, daha iyi şifreleme güvenliği demektir.

Sunulan bir bilimsel çalışmada [56], güvenlik ve performans kriteri olarak MAE değerinin 80' den büyük olması gerektiği belirtilmiştir. Ancak MAE değeri, kaynak görüntünün piksel değerlerine de bağlı olması nedeniyle bu kriter daimi bir koşul olarak gösterilemez. Sunulan kripto sistemde MAE analizi için on farklı test görüntüsü kaynak olarak kullanılmış ve Ş-1 anahtarı kullanılarak bu görüntüler üzerinde şifreleme gerçekleştirilmiştir. Analize ait farklı iterasyon değerlerindeki sonuçlar Tablo 6.9' da verilmiştir.

Tablo 6.9 Şifrelenmiş görüntüler üzerinde MAE analiz sonuçları

Ölçü	**Test görüntüsü**	**MAE**					
		n=1	n=2	n=3	n=4	n=5	**Ortalama**
1024×1024	Dünya	89.640	89.716	89.619	89.728	89.690	89.678
1024×1024	Park	82.822	82.860	82.863	82.786	82.844	82.835
1024×1024	Kedi	77.987	78.097	78.093	78.059	78.040	78.055
512×512	Siyah	127.554	127.427	127.650	127.418	127.724	127.554
512×512	Göl	82.691	82.443	82.616	82.598	82.427	82.555
512×512	Özgürlük Anıtı	75.054	75.035	75.210	75.143	74.975	75.083
256×256	Kameraman	79.229	79.232	79.396	79.392	79.519	79.353
256×256	Ay	106.210	106.648	106.342	106.518	106.237	106.391
256×256	Lena	73.233	72.846	73.057	72.878	72.634	72.929
128×128	Çiçek	83.463	84.002	83.424	83.460	82.459	83.361

Sonuçlardan görülebileceği gibi tüm görüntülere ait MAE sonuçları, farklı iterasyon sayılarında birbirine çok yakındır. Bu durum, analizle ilgili olarak kripto sistemin kararlı bir şifreleme sunduğunu göstermektedir. Ayrıca burada hesaplanan sonuçlar, belirtilen çalışmadaki [56], kriter seviyesine göre büyük veya yakındır.

Sunulan kripto sistemin diferansiyel analizlerinde sağlıklı bir performans değerlendirilmesi yapılabilmesi için elde edilen sonuçlar, mevcut bazı şifreleme algoritmalarıyla karşılaştırılmıştır. Bu analiz ile elde edilen sonuçlar, A5/1 ve W7 akış şifreleyici kullanılmış sonuçlar ile değerlendirilmiştir. A5/1 ve W7, sırasıyla 64-bit ve 128-bit şifreleme anahtarı kullanan senkron şifreleyicilerdir. Her iki algoritma, DES ve AES ile karşılaştırıldığında yeterli güvenlik seviyesine ve performansa sahiptirler. Sunulan şifreleme algoritması ile bazı şifreleme algoritmaları (A5/1, W7, AES) arasındaki şifreleme sonrası performans karşılaştırması Tablo 6.10' da verilmiştir. Burada aynı test görüntüleri kullanılmış olup, sadece tek iterasyon şifreleme gerçekleştirilmiştir.

Tablo 6.10 MAE analiz sonuçlarının diğer algoritmalar ile karşılaştırılması

Ölçü	Test görüntüsü	MAE				
		Sunulan Algoritma		A5/1 [104]	W7 [104]	AES [10]
		Sadece Karıştırma	Karıştırma ve Dağılma			
256×256	Lena	54.440	**72.929**	72.706	72.620	-
256×256	Havaalanı	47.256	**73.380**	72.976	72.762	-
128×128	Tahran	25.842	82.228	-	-	**83.993**

Tablo 6.10' da görüldüğü gibi sunulan kripto sistem, MAE sonuçları açısından A/5 ve W7 şifreleme algoritmalarına göre biraz daha iyi; AES' e göre ise çok yakın seviyededir. Sonuçlar ile ilgili bir diğer önemli nokta ise sadece karıştırma işlemine göre dağılma ve karıştırma işlemlerinin her ikisinin de gerçekleşmesi, MAE sonuçlarını önemli bir şekilde arttırmaktadır. Bu durum, karıştırma ve dağılma işlemlerinin her ikisinin de algoritmada kullanımının iyi bir şifrelemede etkili olduğu sonucunu göstermektedir.

6.2.2 NPCR ve UACI Analizleri

NPCR ve UACI analizleri, bir kripto sistemin diferansiyel saldırılara karşı gösterdiği performansı belirleyen en önemli iki test yöntemidir. Bir görüntü kripto sisteminde, genel olarak kaynak görüntüde bir piksellik değişim, şifrelenmiş görüntüsünde önemli bir değişime neden oluyorsa, böyle bir kripto sistem diferansiyel saldırılara karşı etkin bir şekilde direnecektir.

Sunulan şifreleme yapısı kaynak görüntüye bağlı olacak şekilde tasarlanmıştı ve sistemin kaynak görüntüye olan duyarlılığı, beşinci bölümde detaylıca anlatılmıştı. Dolayısıyla sunulan kripto sistemin, diferansiyel saldırılara karşı etkili olacağı muhtemeldir. Bu analizin ilk yaklaşımı olan NPCR testinde kaynak görüntü P_1, ilk olarak belirli bir şifreleme anahtarıyla şifrelenerek, şifrelenmiş görüntüsü olan C_1 oluşturulur. Daha sonra P_1 görüntüsü üzerinde bir piksellik değişim uygulanarak P_2 görüntüsü elde edilir ve aynı şifreleme anahtarı kullanılarak bu görüntüye karşılık gelen C_2 şifreli görüntüsü oluşturulur. C_1 ve C_2 arasındaki farklı piksel değerlerinin sayısı yüzdelik olarak NPCR yaklaşımıyla ölçülür. Kısaca NPCR, C_1 ve C_2 görüntülerinin birbirinden ne kadar farklı olduğunu belirler ve Denklem (6.7)-(6.8) ile ifade edilir [96].

$$NPCR = \frac{1}{M \times N}\left[\sum_{i=1}^{M}\sum_{j=1}^{N} D(i,j)\right] \times 100\% \tag{6.7}$$

$$D(i,j) = \begin{cases} 0, & if \;\; C_1(i,j) = C_2(i,j) \\ 1, & if \;\; C_1(i,j) \neq C_2(i,j) \end{cases} \tag{6.8}$$

İdeal bir görüntü kripto sisteminde, iki rastgele görüntü için olması gereken NPCR değeri Denklem (6.9)' daki gibi hesaplanır.

$$NPCR_{ideal} = (1 - 2^{-l}) \times 100\% \tag{6.9}$$

Burada *l*, gri görüntüde her bir piksel değerinin temsil edildiği bit sayısıdır. Dolayısıyla NPCR için olması gereken ideal değer, Denklem (6.10)' da verilen

$$NPCR_{ideal} = (1 - 2^{-8}) \times 100\% \tag{6.10}$$

ifadeden $NPCR_{ideal} = \%\,99.60$ olur. UACI ise C_1 ve C_2 görüntüleri arasındaki farkın ortalama yoğunluğunu ölçmek için kullanılır ve Denklem (6.11)' deki gibi tanımlanır [96].

$$UACI = \frac{1}{M \times N}\left[\sum_{i=1}^{M}\sum_{j=1}^{N} \frac{|C_1(i,j) - C_2(i,j)|}{2^l - 1}\right] \times 100\% \tag{6.11}$$

İki rastgele görüntü için ideal UACI değeri, Denklem (6.12)' de verildiği gibi hesaplanır.

$$UACI = \frac{1}{2^{2l}} \cdot \frac{\sum_{i=1}^{2^l - 1} i(i+1)}{2^l - 1} \times 100\% \tag{6.12}$$

Bir gri görüntü için $l = 8$ olduğundan, $UACI_{ideal} = \%\,33.46$ olur. Sonuç olarak, diferansiyel saldırıların etkisiz kalması için görüntü kripto sistemine ait NPCR ve UACI değerleri sırasıyla %99.60 ve %33.46 seviyelerinde olmalıdır. NPCR ve UACI analizleri için kaynak olarak 256×256 ölçüsündeki 'Kameraman' görüntüsü kullanılmıştır. İlk olarak orijinal

'Kameraman' görüntüsü P_1, kripto sistemde Ş-1 şifreleme anahtarı ile şifrelenmiş ve C_1 şifreli görüntüsü elde edilmiştir. Daha sonra kaynak görüntünün $P_1(128,128)$ pozisyonunda olan 154 değeri bir bit değişimle 155 yapılmış ve aynı Ş-1 anahtarı ile yeni görüntü üzerinde tekrar şifreleme gerçekleştirilmiştir. Oluşan C_2 şifrelenmiş görüntü ile C_1 görüntüsü arasındaki NPCR ve UACI sonuçları farklı iterasyon sayıları için Tablo 6.11' de verilmiştir.

Tablo 6.11 'Kameraman' test görüntüsü için NPCR ve UACI değerleri

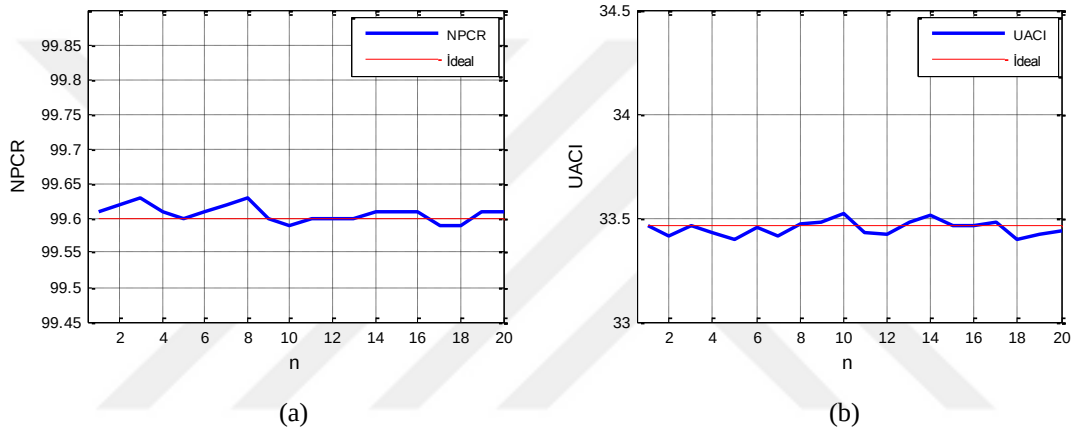
n	**NPCR** (%)	**UACI** (%)
1	99.62	33.56
2	99.63	33.52
3	99.59	33.46
4	99.59	33.49
5	99.56	33.38
6	99.61	33.46
7	99.59	33.54
8	99.60	33.40
9	99.58	33.64
10	99.62	33.48
Ort.	**99.59**	**33.49**

Diferansiyel analiz için şimdi kaynak görüntü olarak çok daha zor bir test görüntüsü seçilsin. Buna göre, 512×512 ölçüsünde ve tüm piksel değerleri sıfır olan 'Siyah' görüntüsü, kripto sistemde kullanılsın. Burada ise kaynak görüntünün $P_1(1,1)=0$ olan ilk pikseli, '1' olarak değiştirilmiş ve oluşturulan yeni görüntü aynı şifreleme anahtarı ile yeniden şifrelenmiştir. 'Siyah' test görüntüsü için hesaplanan NPCR ve UACI değerleri ise Tablo 6.12' de listelenmiştir.

Tablo 6.12 'Siyah' test görüntüsü için NPCR ve UACI değerleri

n	**NPCR** (%)	**UACI** (%)
1	99.61	33.46
2	99.62	33.41
3	99.63	33.46
4	99.61	33.43
5	99.60	33.40
6	99.61	33.45
7	99.62	33.41
8	99.63	33.47
9	99.60	33.48
10	99.59	33.52
Ort.	**99.61**	**33.44**

Analiz sonuçlarından görüldüğü gibi her iki test görüntüsü için hesaplanan sonuçlar, NPCR ve UACI ideal değerlerine çok yakındır ve bu sonuçlar kripto sistemin diferansiyel saldırılara karşı iyi bir yeteneğe sahip olduğunu göstermektedir. Sonuçta, tasarlanan kripto sistem ilk iterasyonda bile kaynak görüntüye karşı tamamen hassas duyarlılık sergilemektedir. Kaynak görüntü tamamen 0 değerlerinden oluşsa bile bu duyarlılık korunmaktadır. Şekil 6.11' de, 'Siyah' görüntüsüne ait NPCR ve UACI sonuçları grafiksel olarak verilmiştir. Grafiksel sonuçlardan da görülebileceği gibi ilk 20 iterasyon için bulunan NPCR ve UACI değerleri, ideal seviyeye çok yakındır.



Şekil 6.11 Kripto sistemin 'Siyah' test görüntüsü için diferansiyel analizi; (a) NPCR, (b) UACI

Kripto sistemin diferansiyel analiz sonuçları A5/1, W7 ve AES algoritmaları ile aynı şartlar altında karşılaştırılmıştır. Karşılaştırılmış sonuçlar Tablo 6.13' de verilmiştir. Bu sonuçlar, açık bir şekilde sunulan kripto sistemin kaynak görüntüye olan hassasiyetinin, diğer algoritmalarınkine göre çok daha yüksek seviyede olduğunu göstermektedir.

Tablo 6.13 Sunulan kripto sistem ile diğer algoritmaların NPCR ve UACI sonuçlarının karşılaştırılması

Ölçü	**Test görüntüsü**	**Diferansiyel Analiz Sonuçları (%)**							
		Sunulan Kripto sistem		A5/1 [104]		W7 [104]		AES [10]	
		NPCR	UACI	NPCR	UACI	NPCR	UACI	NPCR	UACI
256×256	Lena	**99.65**	**33.72**	0.0015	0.0005	0.0015	0.0006	$<10^{-3}$	$<10^{-3}$

6.2.3 MSE ve PSNR Analizleri

Şifrelenmiş bir görüntü orijinal halinden önemli ölçüde farklı olması gerekir. MSE, kaynak ve şifrelenmiş görüntüler arasındaki ortalama karasel hatayı temsil eder ve matematiksel olarak Denklem (6.13)' deki gibi tanımlanır.

$$MSE = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (C(i, j) - P(i, j))^2 \tag{6.13}$$

MSE değeri kripto sistemin şifreleme seviyesi için bir kriter olarak değerlendirilebilir ve büyük MSE değeri, daha iyi bir şifreleme güvenliği demektir. Sunulan bir bilimsel çalışmada [56], şifreleme güvenliği ölçütü olarak MSE değerinin 9555' den büyük olması gerektiği belirtilmiştir. Ancak MSE değeri, şifrelenmiş görüntü dışında kaynak görüntüye de bağlı olması nedeniyle böyle bir kriter şifreleme güvenliği anlamında daimi bir koşul olarak gösterilemez. Ancak bu değerin büyük olması, kaynak görüntü ile buna karşılık gelen şifrelenmiş görüntü arasındaki önemli farklılığın kesinlikle belirtisidir. Farklı ölçülere sahip bazı test görüntülerine ait MSE sonuçları Tablo 6.14' de verilmiştir.

Tablo 6.14 Kripto sistemde kullanılan farklı test görüntüleri için MSE sonuçları

Ölçü	**Test görüntüsü**	**MSE**					
		n=1	n=2	n=3	n=4	n=5	**Ortalama**
1024×1024	Dünya	12034	12039	12022	12047	12038	12036
1024×1024	Park	10274	10288	10290	10278	10283	10282
1024×1024	Kedi	9045	9070	9071	9059	9057	9060
512×512	Siyah	21739	21685	21749	21695	21766	21726
512×512	Göl	10239	10194	10226	10222	10177	10211
512×512	Özgürlük Anıtı	8299	8291	8323	8307	8292	8302
256×256	Kameraman	9409	9364	9400	9413	9434	9404
256×256	Ay	16254	16389	16277	16355	16284	16312
256×256	Lena	7807	7722	7754	7752	7698	7746
128×128	Çiçek	7734	7741	7715	7629	7687	7701

Burada kullanılan test görüntülerine ait MSE sonuçları, farklı iterasyon sayılarında birbirine yakın olup, kripto sistemin kararlı bir şifreleme gerçekleştirdiğini göstermektedir.

PSNR analizi ise kripto sistemin şifreleme kalitesi ile ilişkili bir yöntemdir ve bu değer ne kadar büyükse şifreli görüntü kaynak görüntüye o kadar yakındır. Dolayısıyla, bir kripto sistem için daha küçük bir PSNR değeri, daha iyi bir şifreleme yapısı demektir. PSNR parametresi Denklem (6.14)' de verildiği gibi desibel olarak hesaplanır.

$$PSNR = 20.\log_{10}\left[P_{\max} / \sqrt{MSE} \right] \tag{6.14}$$

Burada $P_{\max}$, görüntü içerisindeki bir pikselin alabileceği en büyük değerdir yani $P_{\max} = 255$ dir. Denklem (6.14)' den görüldüğü gibi PSNR, doğrudan MSE değerine bağlıdır. Sunulan kripto sistemin PSNR analizi için Tablo 6.15' de verilen test görüntüleri kullanılmış ve analiz sonuçları listelenmiştir. Burada MSE değeri olarak Tablo 6.14' de hesaplanan ortalama değerler referans alınmıştır.

Tablo 6.15 PSNR analiz sonuçları

Ölçü	Test görüntüsü	PSNR (dB)
1024×1024	Dünya	7.3260
1024×1024	Park	8.0100
1024×1024	Kedi	8.5595
512×512	Siyah	4.7610
512×512	Göl	8.0401
512×512	Özgürlük Anıtı	8.9389
256×256	Kameraman	8.3976
256×256	Ay	6.0057
256×256	Lena	9.2400
128×128	Çiçek	9.2653

6.2.4 Şifreleme Kalitesi Analizi

Şifreleme sonrası piksel değerleri, orijinal görüntüsündekinden çok farklı ve düzensiz bir şekilde oluşması gerekir. Piksel değerlerindeki değişme ne kadar yüksek ise görüntü şifreleme o kadar etkili olur. Bu değişme şifreleme kalitesiyle doğrudan ilişkilidir. Şifreleme kalitesi (EQ), kaynak görüntü ile şifrelenmiş görüntü arasındaki piksel değerlerindeki toplam değişim olarak ifade edilir ve Denklem (6.15)' deki gibi tanımlanır.

$$EQ = \frac{\sum_{L=0}^{255} \left| H_L(C) - H_L(P) \right|}{256} \tag{6.15}$$

Burada $H_L(C)$ ve $H_L(P)$, sırasıyla şifrelenmiş görüntüde ve kaynak görüntüde her bir piksel değerinin oluşma sayısını gösterir. Kısaca, EQ parametresi her bir piksel değer seviyesi için ortalama değişim sayısını temsil eder. Şifreleme kalitesi anlamında EQ değerinin büyük olması önemlidir. Sunulan bir bilimsel çalışmada [56], bu değerinin 210'

dan büyük olması gerektiği belirtilmiştir. Şifreleme kalitesi analizi için farklı ölçülerdeki bazı test görüntüleri kripto sistemde kullanılmış ve sonuçlar Tablo 6.16' da verilmiştir. Ayrıca sunulan kripto sistemin şifreleme kalitesi performansı, aynı kaynak görüntünün kullanıldığı diğer şifreleme algoritmaları ile karşılaştırılmıştır ve sonuçlar Tablo 6.17' de verilmiştir.

Tablo 6.16 Sunulan kripto sistemin şifreleme kalitesi analizi

Ölçü	**Test görüntüsü**	**Şifreleme Kalitesi (EQ)**					
		n=1	n=2	n=3	n=4	n=5	**Ortalama**
1024×1024	Dünya	4556	4558	4551	4552	4559	4555
1024×1024	Park	3456	3463	3454	3450	3452	3455
1024×1024	Kedi	3258	3262	3268	3265	3263	3263
512×512	Siyah	2039	2040	2041	2039	2040	2039
512×512	Göl	582	576	576	577	578	577
512×512	Özgürlük Anıtı	626	630	632	631	630	629
256×256	Kameraman	240	241	242	242	240	241
256×256	Ay	309	309	310	308	309	309
256×256	Lena	164	165	163	164	163	163
128×128	Çiçek	47	48	47	48	47	47

Tablo 6.17 Sunulan kripto sistem ile diğer algoritmaların EQ sonuçlarının karşılaştırılması

Test görüntüsü	**Şifreleme Kalitesi (EQ)**		
	Sunulan Kripto sistem	A5/1 [104]	W7 [104]
Havaalanı	**291.43**	289.16	288.08
Adam	**147.59**	145.09	143.71
Köprü	141.09	**141.32**	140.84
Tesis	**208.96**	207.10	206.14
Lena	164.41	**169.38**	168.50
Grafik	425.63	**455.33**	454.61
Çift	**316.04**	222.86	220.66
Saat	241.87	**242.33**	241.59

Kripto sistemin diğer analiz sonuçlarında olduğu gibi buradaki analizde de EQ sonuçlarının dengeli olduğu açıktır. Tablo 6.17' deki sonuçlara göre sunulan kripto sistemin şifreleme kalitesi performansı, A5/1 ve W7 şifreleme algoritmalarına göre daha iyi seviyededir. Karşılaştırma için kullanılan 256×256 ölçülerine sahip test görüntüleri USC-SIPI görüntü veri tabanı (Güney Kaliforniya Üniversitesi) sitesinden elde edilmiştir.

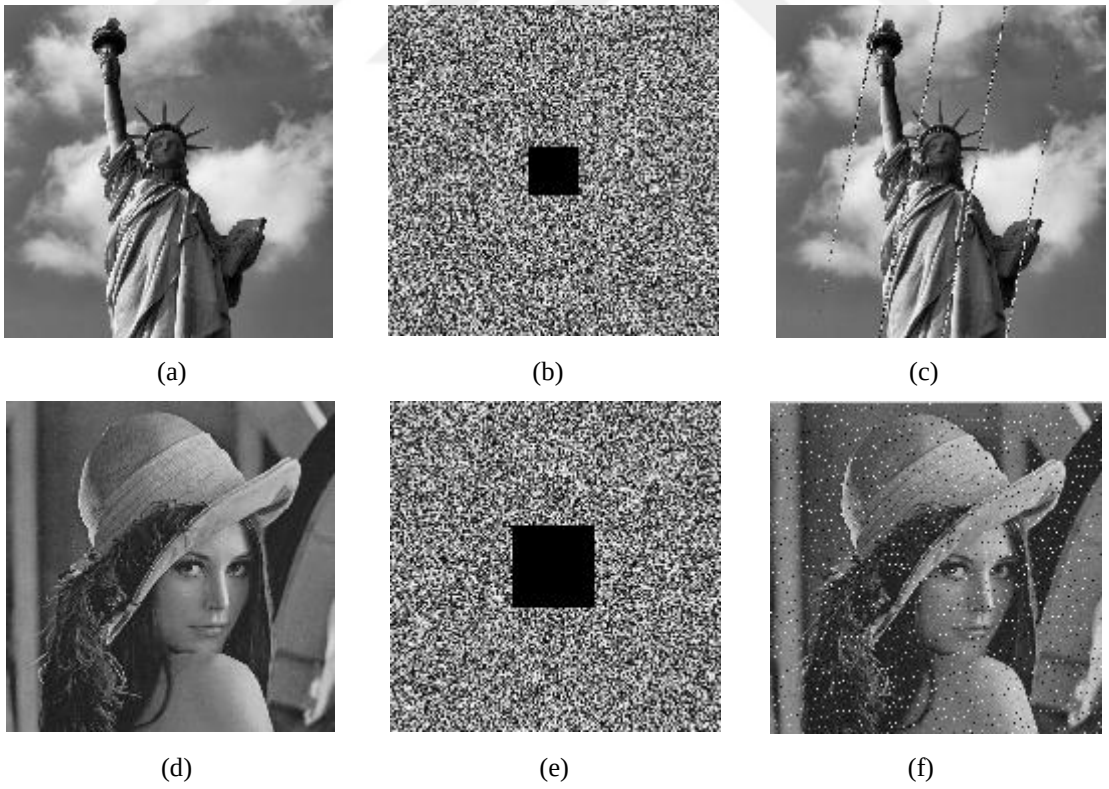
İstatistiksel ve diferansiyel analiz yöntemleri içerisinde hesaplanan matematiksel ve grafiksel sonuçlarla beraber şifreleme ve çözme işlemleriyle ilgili gerçekleştirilen deneysel benzetimler, tasarlanan kripto sistemin güvenlik yönünden tatmin edici bir performansa

sahip olduğunu göstermektedir. Çalışmanın bundan sonraki kısmında ise sunulan kripto sistemin şifreleme ve çözme hızı incelenecektir. Ayrıca sistemin gürültü ve veri kaybı altındaki performansı ele alınacaktır.

6.3. Kripto Sistemin Performans Analizi

6.3.1 Veri Kaybı ve Gürültü Saldırısı Analizleri

Güçlü bir görüntü şifreleme sistemi, iletim sırasında oluşabilecek veri kaybına karşı dirençli olması gerekir. Şifrelenmiş görüntüye ait çözümlenme işleminin ciddi oranda etkilenmemesi için bu özelliğin kripto sistemde sağlanması önemlidir. Sunulan kripto sistemin veri kaybı analizi bilgisayar ortamında Matlab yazılımı ile gerçekleştirilmiş ve bunun için farklı bir çok test görüntüsü kullanılmıştır. Elde edilen görsel analiz sonuçları birbirleriyle benzerlik gösterdiğinden dolayı burada sadece iki görüntüye ait veri kaybı analiz sonucuna yer verilmiştir. Şekil 6.12, iki farklı şifrelenmiş test görüntüsüne uygulanan veri kaybı analiz sonucunu göstermektedir.



(a) (b) (c)

(d) (e) (f)

Şekil 6.12 (a) 'Özgürlük Anıtı' görüntüsü, (b) Şifrelenmiş 'Özgürlük Anıtı' görüntüsünde 75×75 büyüklüğünde veri kaybı, (c) Çözülmüş 'Özgürlük Anıtı' görüntüsü, (d) 'Lena' görüntüsü, (e) Şifrelenmiş 'Lena' görüntüsünde 125×125 büyüklüğünde veri kaybı, (f) Çözülmüş 'Lena' görüntüsü

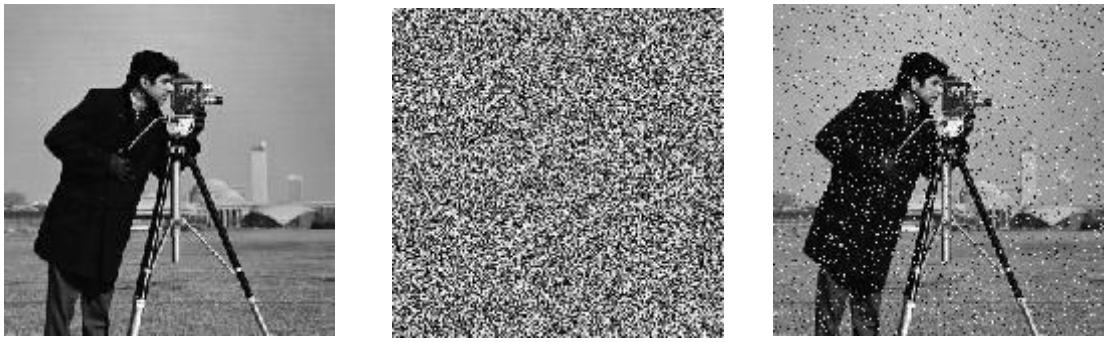
İlk olarak rastgele seçilmiş test görüntülerinden 512×512 ölçüsüne sahip 'Özgürlük Anıtı' görüntüsü, sunulan algoritma ile tek iterasyon kullanılarak şifrelenmiştir. Daha sonra elde edilen şifrelenmiş görüntünün orta bölgesinde 75×75 büyüklüğünde veri kaybı gerçekleştirilmiş ve oluşan görüntü üzerinde çözme işlemi uygulanmıştır. Aynı işlem bu sefer 'Lena' görüntüsü üzerinde iki şifreleme iterasyonu kullanarak 125×125 ölçülerinde veri kaybıyla gerçekleştirilmiştir. Bu işlemlere ait görsel sonuçlar Şekil 6.12' de verilmiştir.

Şifrelenmiş görüntülere farklı büyüklükteki veri kaybı uygulanarak elde edilen veri kayıplı şifrelenmiş görüntülerin çözümlenmesiyle oluşan görüntüler ile kaynak görüntüleri arasındaki benzerlik katsayıları hesaplanmış ve sonuçlar Tablo 6.18' de verilmiştir. Bu sonuçlar, şifrelenmiş görüntüdeki veri kaybı %24 oranında olsa bile çözümlenmiş görüntünün anlamlı olduğunu ve kripto sistemin veri kaybına karşı dayanıklı olduğunu göstermektedir. Çünkü bu şartlar altında kurtarılan görüntü, orijinal görüntüye yüksek oranda benzemektedir. Yüksek oranda görüntünün çözümlenmesinde, kripto sistemde kullanılan karıştırma kısmının rolünün büyük olduğu söylenebilir.

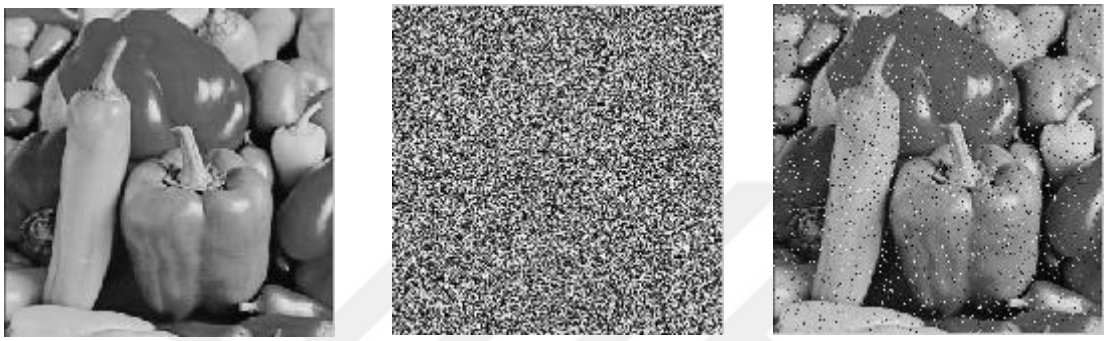
Tablo 6.18 Sunulan kripto sistemin veri kaybı analizi

Test görüntüsü	İterasyon	Veri Kaybı Ölçüsü	Veri Kaybı (%)	Benzerlik Katsayısı
Özgürlük Anıtı	1	75×75	2.14	0.9641
Lena	2	125×125	5.96	0.8742
Babun	3	150×150	8.58	0.7103
Göl	2	250×250	23.84	0.6601

Gürültü saldırısı analizi için kripto sistemden elde edilmiş 256×256 ölçüsüne sahip bazı şifrelenmiş görüntülere, farklı gürültü yoğunluğunda 'Tuz & Biber' gürültüsü ilave edilmiştir. Genellikle bu gürültü, görüntü iletimi sırasında bozulmuş piksellerin yüzdesi ile nitelendirilir. Örneğin, 0.01 gürültü yoğunluğuna sahip 'Tuz & Biber' gürültüsü ilave edilmiş bir görüntü %1 oranında bozulmuş pikseller içerir. Burada bozulmuş piksel, piksel değerlerinin sıfıra (Siyah) ya da maksimum değere (Beyaz) ayarlanması anlamındadır. Sırasıyla 0.05 ve 0.02 gürültü yoğunluğu ilave edilmiş 'Kameraman' ve 'Biber' görüntüleri sunulan kripto sistem tarafından çözülmeye çalışılmıştır. Kripto sistemin gürültü saldırısı analizi ile ilgili sonuçlar, Şekil 6.13' de gösterilmiştir.



(a) (b) (c)



(d) (e) (f)

Şekil 6.13 (a) 'Kameraman' görüntüsü, (b) Şifrelenmiş 'Kameraman' görüntüsüne 0.05 oranında gürültü yoğunluğu eklenmiş görüntü, (c) Çözülmüş 'Kameraman' görüntüsü, (d) 'Biber' görüntüsü, (e) Şifrelenmiş 'Biber' görüntüsüne 0.02 oranında gürültü yoğunluğu eklenmiş görüntü, (f) Çözülmüş 'Biber' görüntüsü

Gürültü eklenmiş şifreli görüntülerin çözümlenmesi ile kurtarılan görüntü yüksek oranda orijinal görüntüye benzemektedir. Bu durum, sadece görsel sonuçlardan değil aynı zamanda Tablo 6.19' da verilen benzerlik analiz sonuçlarından da görülebilmektedir. Dolayısıyla, sunulan kripto sistemin gürültü saldırısına karşı başarılı olduğu söylenebilir.

Tablo 6.19 Sunulan kripto sistemin gürültü saldırısı analizi

Test görüntüsü	İterasyon	Gürültü (%)	Benzerlik Katsayısı
Kameraman	1	5	0.8794
Biber	2	2	0.8920
Ay	3	1	0.8477

Gürültülü görüntünün kurtarılması kısmında şifrelemedeki iterasyon sayısının çözümlemeye etkisini gözlemlemek için aynı görüntüler sabit gürültü yoğunluğu seviyesinde de ele alınmıştır. Bu analize ait sonuçlar ise Tablo 6.20' de verilmiştir. Burada şifrelenmiş görüntünün çözümlenmesinde orijinal görüntü ile benzerlik katsayısı referans

alınmıştır ve gürültü yoğunluğu olarak Matlab programında varsayılan değer olan 0.05 kullanılmıştır.

Tablo 6.20 Gürültü saldırısı analizinde iterasyon sayısının çözümlemeye etkisi

Test görüntüsü	iterasyon sayısı				
	n=1	n=2	n=3	n=4	n=5
Kameraman	0.8794	0.7880	0.6347	0.3919	0.1624
Biber	0.8595	0.7587	0.5727	0.3557	0.1396
Ay	0.8183	0.6943	0.5261	0.3288	0.1442

Tablo 6.20' de verilen sonuçlar, iterasyon sayısının gürültü ilave edilmiş şifreli görüntünün çözümlenmesini açık bir şekilde etkilediği görülmektedir. Burada iterasyon sayısı arttıkça sabit gürültü altında orijinal görüntünün kurtarılması güçleşmektedir. Bu durum aynı zamanda gürültü yoğunluğuna da bağlıdır. Gürültüden kaynaklı çözme hatasının her geçen iterasyonda artması, böyle bir sonucu olağan yapmaktadır.

6.3.2 Şifreleme ve Çözme Hızı Analizi

Sunulan kripto sistemin şifreleme ve çözme hızlarını değerlendirebilmek için yeterli sayıda test görüntüsü bu amaç için kullanılmıştır. Intel i7 3.4 Ghz işlemci ve 4GB RAM kullanan bir bilgisayarda farklı ölçülerdeki test görüntüleri, sunulan kripto sistemin şifreleme ve çözme hızlarının tespitinde kullanılmıştır. Sonuçlarla ilgili ortalama hesaplama süresi Tablo 6.21' de verilmiştir. Burada her bir görüntü ölçüsü için on farklı test görüntüsü kullanılmış ve iterasyon sayısı olarak 1 seçilmiştir.

Tablo 6.21 Sunulan algoritmanın şifreleme ve çözme hızı analiz sonuçları

Test Görüntüsü ölçüsü	Ortalama Karıştırma & Dağılma toplam süresi (sn)	Çözme Süresi (sn)	Şifreleme Oranı (Mbps)	Çözme Oranı (Mbps)
128×128	0.02	0.02	6.24	5.69
256×256	0.08	0.09	6.47	5.76
512×512	0.32	0.34	6.55	6.17
1024×1024	1.29	1.40	6.50	5.99
2048×2048	5.31	5.68	6.32	5.91

Sunulan kripto sistemin hız performansı, A5/1 ve W7 algoritmalarıyla karşılaştırılmıştır. Sunulan çalışmada [104], A5/1 ve W7 algoritmalarının Intel Core 2 Duo 2.1 Ghz işlemci ve 2 GB RAM kullanan bir bilgisayarda gerçekleştirilen hız testi verilmiştir. Tablo 6.22' de sunulan kripto sistemin ve diğer şifreleme algoritmalarının şifreleme hızları karşılaştırılmıştır.

Tablo 6.22 Kripto sistemin diğer algoritmalarla hız performansının karşılaştırılması

Şifreleme hızı (sn)		
Görüntü ölçüsü	128×128	256×256
AES-128 [104]	10.12	65.23
A5/1 [104]	0.87	2.32
W7 [104]	1.01	3.23
Sunulan algoritma	**0.02**	**0.08**

6.4. Kaba-Kuvvet Saldırı Tekniğinin Sunulan Algoritmaya Uygulanışı

Kriptografik bir saldırı bir kripto sistemin güvenliğini aşmak için kullanılan bir yöntem olup, şifreleme algoritması veya anahtar yönetim düzeninde bir zayıflık bulma amacı taşır. Bu kısımda, gizli anahtar kodlarının bulunmasına yönelik gerçekleştirilen kaba-kuvvet saldırı yönteminin sunulan algoritmaya uygulanışı ile ilgili değerlendirme sonuçları ele alınacaktır. Bu saldırı yönteminde olası tüm şifreleme anahtarları sistematik bir şekilde doğru anahtar bulunana kadar denenir. Örneğin, anahtar alanı *n* bit olan bir kripto sistemde saldırgan algoritmayı kırmak için ortalama 2^{n-1} sayıda anahtar dener [105]. Şifreleme işleminde kullanılan anahtar uzunluğu bu saldırı tekniğinin pratik olarak uygulanabilirliğini belirler. Dolayısıyla bu yöntem ile büyük anahtar uzunluğuna sahip kripto sistemleri, küçük olanlarına göre kırmak çok daha zordur [106].

Kripto sistemi kırma maliyetinin şifreli bilginin değerini ya da şifreleyici yapısını kırmak için gerekli süre yararlı bilginin ömrünü aşıyor ise bu şifreleme yapısı hesaplamaya dayalı güvenlidir. Aksi durumda tüm kripto sistemler yeterli kaynak ve zaman içerisinde kaba-kuvvet saldırı tekniğiyle kırılması mümkündür. Bir şifreleme sisteminin bu yönteme karşı dayanıklılığı teorik olarak başarılı bir kaba-kuvvet saldırısının ne kadar zaman aldığına bağlıdır. Burada kaba-kuvvet saldırısı için gerekli kaynaklar (zaman, bellek gibi) anahtar aralığı artışıyla doğrusal olarak değil, üstel olarak artar [106].

Kaba-kuvvet saldırı yönteminin, sunulan kripto sisteme uygulandığı ele alınsın. Kripto sistemde K_1 ve K_2 gizli kodlar olup, şifreleme anahtarları olarak s_1, s_2, x_0, x_0'

parametrelerinden elde edilmekteydi. Ayrıca kripto sistemde kullanılan algoritma, K_1 ve K_2 kodları her bir iterasyonda tamamen değişecek şekilde tasarlanmıştı. Burada K_1 veya K_2, 0 ile 255 arasında yani 8-bit uzunluğunda bir değerdir. Dağılma fonksiyonunda şifrelenmiş piksel bir önceki şifreli piksel değerine bağlı olduğundan dolayı şifreleme boyunca elde edilen tüm şifreli pikseller de, $c(0)$ olarak tanımlanan ilk başlangıç değerine bağlı olacaktır. Bu anlamda, $c(0)$ değeri şifreleme işlemini belirleyen 8-bit uzunluğunda gizli bir başlangıç kodu olarak da düşünülebilir.

Kripto sistemin karıştırma kısmında ise p ve q anahtar parametreleri kullanılmıştı. Bu parametrelerin şifrelenecek olan görüntünün ölçüsü seviyesinde sınırlanması gerektiği, kripto sistemin anahtar alanı analizi kısmında değinilmişti. Sonuç olarak bir tek pikselin *n* iterasyon ile şifrelenmesinde kullanılabilecek muhtemel gizli kod sayısı, Denklem (6.16)' da verildiği gibi oluşacaktır.

$$B = \left[K_1.K_2\right]^n .c_{Baslan}.p.q = \left[2^8.2^8\right]^n .2^8.N.N = 2^{8(2n+1)}.N^2 \tag{6.16}$$

Ölçüsü $N \times N$ olan bir görüntüde N^2 kadar piksel olacağından dolayı kaba-kuvvet saldırı tekniği için denenmesi gereken muhtemel kodların toplam sayısı Denklem (6.17)' deki gibi olacaktır.

$$B = 2^{8(2n+1)}.N^4 \tag{6.17}$$

Görüldüğü üzere bu değer kaynak görüntünün büyüklüğüne ve iterasyon sayısına bağlıdır. En kötü ihtimalle Denklem (6.17)' de verilen sayı kadar deneme gerçekleştirildiğinde gizli kodların bulunacağı kesindir. Ancak genellikle bunun yarısı kadar kombinasyonun denenmesi, kaba-kuvvet saldırısı için yeterli olduğu kabul edilir. Sonuç olarak, bu saldırı yönteminin pratiksel anlamda uygulanabilirliğinin denetlenmesi için Denklem (6.18)' de verilen ifade referans alınmalıdır.

$$B = \frac{2^{8(2n+1)}.N^4}{2} \tag{6.18}$$

Örneğin, 512×512 ölçüsünde bir test görüntüsü ele alınsın. Bu görüntü sunulan kripto sistemde herhangi bir şifreleme anahtarıyla $n=3$ iterasyon sayısıyla şifrelensin. Buna göre kaba-kuvvet saldırı için denenmesi gereken toplam kod sayısı, Denklem (6.19)' da hesaplandığı gibi olacaktır.

$$B=\frac{2^{8(2n+1)}.N^4}{2}=\frac{2^{8(2.3+1)}.(2^9)^4}{2}=2^{91} \tag{6.19}$$

Eğer saldırgan bilgisayar işlemcisi hızı ölçümü olarak, saniye başına komut sayısı 177,730 M (Instructions Per Seconds, IPS) olan bir işlemci (Intel Core I7) kullanarak verilen örneğe kaba-kuvvet tekniğiyle saldırırsa, bilgisayarın hesaplama yükü yıl olarak,

$$\frac{2^{91}}{177730\times10^6\times60\times60\times24\times365}=4.41\times10^8 \tag{6.20}$$

olur. Buna göre, Denklem (6.20)' de hesaplanan bu sonuç çok uzun bir zaman alacağından, kaba-kuvvet saldırısı pratiksel anlamda başarısız olur. 56-bit uzunluğundaki DES algoritması ise böyle bir donanımsal kaynak altında kaba-kuvvet yöntemiyle yaklaşık 4 günde kırılmaktadır. Farklı ölçülere sahip görüntülerde ve şifreleme iterasyonunda kripto sisteme uygulanan kaba-kuvvet saldırısının başarılı olabilmesi için gerekli zaman Tablo 6.23' de verilmiştir.

Tablo 6.23' de hesaplanan süreler uygulama noktasında dikkate alındığında, kaba-kuvvet saldırısının başarısız olabilmesi için en az üç defa şifrelemenin gerçekleştirilmesi gerektiği anlaşılmaktadır. Kaos tabanlı şifreleme sistemlerinin yüksek güvenlik sağlayabilmeleri için şifreleme iterasyonunun birkaç kez tekrarlanması gerektiği mevcut çalışmalarda da ifade edilmiştir.

Tablo 6.23 Kaba-kuvvet saldırısının sunulan algoritmaya uygulanışı

Görüntü Ölçüsü	İterasyon	B	Süre
128×128	$n=1$	2^{51}	3.5 saat
	$n=2$	2^{67}	26.3 yıl
	$n=3$	2^{83}	1.7×10^{6} yıl
	$n=4$	2^{99}	1.1×10^{11} yıl
256×256	$n=1$	2^{55}	2.4 gün
	$n=2$	2^{71}	421 yıl
	$n=3$	2^{87}	2.8×10^{7} yıl
	$n=4$	2^{103}	1.8×10^{12} yıl
512×512	$n=1$	2^{59}	38 gün
	$n=2$	2^{75}	6.7×10^{3} yıl
	$n=3$	2^{91}	4.4×10^{8} yıl
	$n=4$	2^{107}	2.9×10^{13} yıl
1024×1024	$n=1$	2^{63}	1.6 yıl
	$n=2$	2^{79}	10^{5} yıl
	$n=3$	2^{95}	7×10^{9} yıl
	$n=4$	2^{111}	4.6×10^{14} yıl
2048×2048	$n=1$	2^{67}	26.3 yıl
	$n=2$	2^{83}	1.7×10^{6} yıl
	$n=3$	2^{99}	1.1×10^{11} yıl
	$n=4$	2^{115}	7.4×10^{15} yıl

6.5. Çığ Etkisi Analizi

Çığ etkisi (Avalanche Effect) analizi, herhangi bir kripto sistemin kaynak görüntüye olan direncini ölçmek için kullanılan bir değerlendirme kriteridir. Kripto sistemin kaynak görüntü üzerindeki değişim hassasiyetine göre verdiği tepki bu yöntem ile belirlenebilir. Çığ etkisi bilgi teorisinde kullanılan Hamming Mesafesi (Hamming Distance, HD) ile ölçülür. Bu parametre, eşit uzunluktaki iki seri için karşılıklı pozisyonlardaki farklı sembollerin sayısıdır ve Denklem (6.21)' deki gibi tanımlanır.

$$HD(C_1, C_2) = \frac{1}{|I_b|}\sum_{K=1}^{|I_b|}(C_1(K) \oplus C_2(K)) \tag{6.21}$$

Burada, $|I_b| = M \times N \times 8$ bit formatında olan görüntü ölçüsü uzunluğunu gösterir. C_1 ve C_2 ise sırasıyla kaynak görüntüde bir bit değişimle oluşturulmuş şifrelenmiş görüntüleri göstermektedir. Burada ideal HD değeri %50 dir [21]. İyi bir şifreleyici, %50' ye yakın bir

HD değer üretir. Yani kaynak görüntüdeki bir bit değişim, buna karşılık gelen şifrelenmiş görüntüdeki her bir biti, ½ olasılıkla değiştirecek demektir. Dolayısıyla kaynak görüntü hassasiyeti tabanlı saldırılar bu durumda kullanışsız hale gelecektir.

Çığ etkisi analizi için farklı ölçülerdeki test görüntüleri sunulan kripto sisteme uygulanmıştır. Burada test görüntülerin birinci piksel değerleri sadece bir bit değiştirilerek rastgele seçilmiş bir şifreleme anahtarıyla tek iterasyon sonunda şifrelenmiş görüntüleri elde edilmiştir. Bu analize ait HD değerleri, Tablo 6.24' de verilmiştir.

Tablo 6.24 Çığ Etkisi analiz sonuçları

Görüntü ölçüsü	**HD** (%)
128×128	49.6505
256×256	49.9879
512×512	49.9455
1024×1024	50.0104
2048×2048	50.0226

Tablo 6.24' de verilen analiz sonuçlarına göre hesaplanan HD değerleri, ideal değere çok yakındır. Bu sonuçlar, bir kez daha göstermektedir ki sunulan kripto sistem kaynak görüntüye karşı aşırı duyarlıdır ve kaynak görüntüde bir bit değişimde bile karşılık gelen şifrelenmiş görüntüde ki her bir bit değeri yaklaşık %50 olasılıkla değişmektedir.

7. SUNULAN ŞİFRELEME ALGORİTMASININ DONANIMSAL ORTAMDA GERÇEKLEŞTİRİLMESİ

Bu kısımda tasarlanan kripto sistem uygulamalı olarak sahada programlanabilen kapı dizileri (FPGA) ortamında gerçekleştirilecektir. Bu kısımda FPGA uygulamaları ile ilgili sonuçlar iki ana kısımda ele alınmıştır. İlk olarak İLH üretecinin FPGA platformunda gerçek zamanlı olarak çalıştırılması ile ilgili sonuçlara yer verilecek ve daha sonra tasarlanan kripto sistemin donanımsal benzetimi ile birlikte FPGA donanımının, video grafik dizisi (VGA) portu üzerinden görüntünün aktarılmasına yönelik sonuçlar paylaşılacaktır. Burada gerçekleştirilen her iki yaklaşıma ait tasarımlar, MATLAB yazılımı kullanılarak Simulink ortamında modellenmiştir. Sunulan görüntü kripto sisteminde, özellikle sisteme ait alt birimlerin modellenmesinde MATLAB yazılımı; donanım tabanlı uygulama ve benzetim işlemlerinde ise FPGA yoğun olarak kullanılmıştır.

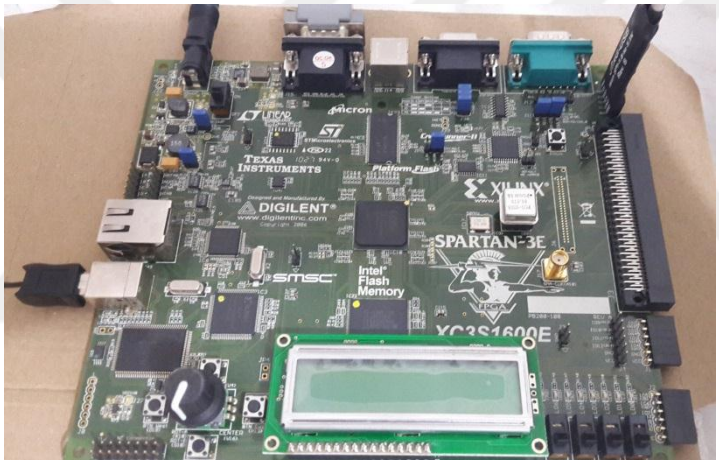
7.1. FPGA Mimarisi

Günümüz teknolojisinin hızla gelişmesiyle birlikte kullanılan elektronik devreler daha karmaşık hale gelmiş ve bunun sonucunda da bilinen yöntemlerle bu devreleri tasarlamak oldukça zor bir hal almıştır. Bu ihtiyacı karşılamak için yeni teknolojiler geliştirilmekle beraber bunlardan biri de FPGA mimarisinin kullanıldığı teknolojidir. FPGA bir dizi yapılandırılabilir mantık blokları içeren, mimari anlamda silikon bir yonga olmakla beraber istenen fonksiyona göre kullanıcı tarafından sahada programlanabilen bir devredir. Sahada programlanabilir isminin verilmesinin nedeni mantık bloklarının ve ara bağlantı noktalarının üretim sürecinden sonra uygulama alanına yönelik olarak programlanabilmesidir. Dolayısıyla, bu sistemin donanım yapısı kullanıcı tarafından yeniden değiştirilebilir ve tekrar programlanabilir [107].

FPGA sistemleri belirtilen yapılandırma verilerini (programlama dosyası), dahili bellek hücrelerine yükleyerek özelleştirilir. Bu mimari yapı, programlanabilir mantık blokları (Configurable Logic Block, CLB), bu blok dizisini çevreleyen giriş/çıkış blokları (Input/Output Blocks, I/OB) ve ara bağlantılar olmak üzere düzenlenebilir üç ana bölümden oluşur. FPGA sistemleri özellikle paralel işlem gerektiren uygulamalarda yaygın olarak kullanılmaktadır. Özellikle tasarım sırasında kullanıcıya esneklik sağlaması sebebiyle kullanımı gittikçe yaygınlaşmıştır. FPGA bileşenlerinden lojik bloklar, mantık

kapılarıyla gerçekleştirilebilen basit işlemlerin yapılması için programlanabildikleri gibi, şifre çözücü seviyesinde daha karmaşık matematiksel fonksiyonlara ait işlemleri de yerine getirebilir. Savunma sanayisi, sayısal işaret işleme, telekomünikasyon, otomotiv, FPGA sistemlerinin uygulama alanlarından bazılarıdır.

Bu çalışmada gerçek zamanlı uygulama ve donanımsal benzetim işlemleri için Xilinx firmasına ait Spartan 3E-XC3S1600E sınıfı bir FPGA kullanılmıştır. Kullanılan FPGA teknik özellik olarak: 33,192 mantık hücresi, 648 Kbits hızlı blok RAM, verimli olarak dağıtılmış 231 Kbits RAM, 3,688 programlanabilir mantık blokları, 376 maksimum I/O ve kart üzerinde yerleşik 50 Mhz bir osilatör içermektedir. Şekil 7.1' de bu çalışmada kullanılan FPGA gösterilmiştir.



Şekil 7.1 Uygulama çalışmasında kullanılan FPGA donanımı

FPGA sistemleri donanım tanımlama dilleri ile programlanırlar. Donanım tanımlama dili, fonksiyonu gerçekleştirecek birimin nasıl çalışması gerektiğinin tanımlanması için kullanılır. FPGA, yaygın olarak kullanılan standart iki donanım tanımlama dili, VHDL (Very high speed integrated circuit Hardware Description Language) veya Verilog (Verifying Logic) ile programlanır. Bu diller, bilinen standart bilgisayar programlama dilleri gibi düşünülmemelidir. Örneğin, C bilgisayar dili çalışma mantığı açısından sıralı komutlarla çalışırken; VHDL dili ise esas olarak paralel mantıkla işlev gördüğü gibi sıralı komutları da destekler. VHDL, sayısal devreleri tanımlamak için kullanılan üst düzey seviyeli ve yüksek hızı destekleyen bir programlama dili olmasının yanında donanıma ait davranışı, sistem seviyesinden lojik kapı seviyesine nitelendirebilmeyi de mümkün

kılmaktadır. Bu çalışmada FPGA donanımını programlamak için koda dayalı kısımlarda VHDL dili tercih edilmiştir.

7.2. Xilinx ISE Yazılımı

Bu çalışmanın uygulama tarafıyla ilgili olarak FPGA donanımın programlanmasında, sentezlenmiş tasarımlara ait giriş/çıkış birimlerinin belirlenmesinde, HDL benzetim sonuçlarının gözlemlenmesinde, 'bit' uzantılı programlama dosyasının FPGA sistemine yüklenmesinde, FPGA bellek birimine veri yazılımda ve erişiminde, Xilinx ISE (Integrated Synthesis Environment) yazılımı kullanılmıştır. Xilinx ISE, Xilinx firması tarafından HDL tasarımlarının sentezi ve analizi için geliştirilmiş olup, geliştiricinin tasarımlarını sentezleyebilmesine (derlemesine), tasarımların tepkisini benzetebilmesine ve hedef cihazın programlanabilmesine olanak sağlayan üst düzey seviyeli bir yazılım aracıdır.

Xilinx Sistem Üreteci (Xilinx System Generator, XSG) ise MATLAB/Simulink tabanlı çalışan ve Xilinx marka FPGA sistemleri için geliştirilmiş bir tasarım aracıdır. Bu sayede, model tabanlı Simulink ortamının kullanımı, Xilinx blokları ile oluşturulan FPGA tasarımları için mümkün hale gelir ve bu tasarımların sentezlenmesi, donanımsal benzetimlerinin (hardware co-simulation) gerçekleştirilmesi sağlanır. FPGA donanımının programlanması için sentezlenebilen VHDL kodu da XSG tarafından üretilir. Bunun yanında XSG, donanımsal benzetim adı verilen, gerçek donanımı benzetime getirerek FPGA ortamında çalışan bir tasarımın doğrudan Simulink ortamındaki benzetime dahil edilmesini gerçekleştirebilme yeteneğine sahiptir. Donanım benzetimi, tasarım üzerinden hedef cihazın programlanabileceği bir bit akışını otomatik olarak oluşturur ve ilgili tasarımı bir blok ile ilişkilendirir. Böylelikle, tasarım Simulink ortamında benzetildiğinde, sentezlenmiş kısım mevcut FPGA donanımında işlenirken veya hesaplanırken sonuçlar ise Simulink ortamında gözlemlenebilmektedir. Karmaşık bir tasarım benzetiminin hızlandırılmasında ve tasarımın gerçekte donanım içerisinde çalıştığının doğrulanmasında donanımsal benzetim kullanılabilir [108].

7.3. İLH Üretecinin Modellenmesi ve FPGA Ortamında Gerçekleştirilmesi

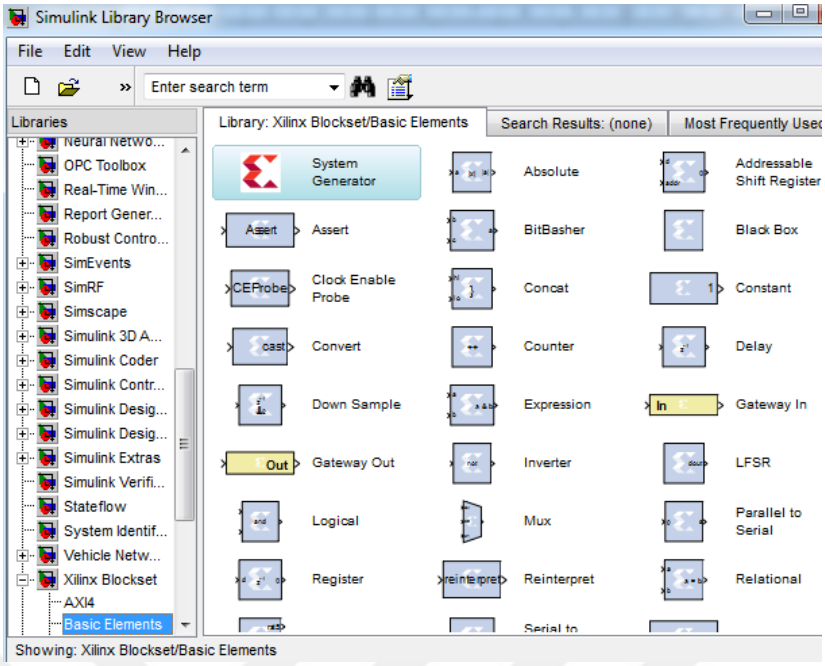
Anahtarlamalı kondansatör veya analog CMOS teknolojisi gibi analog devre tabanlı birçok yöntem, kaotik üreteç gerçekleştirmek için kullanılabilir. Fakat bu yöntemler, devre bileşenlerinin değeri, kullanılan elemanların ömürleri veya sıcaklık gibi etkenlerle değişebildiğinden, uygulamalarda bazı pratik güçlükler sergilerler. Bu anlamda analog

gerçekleştirme zor olmakla beraber belli ölçülerle bu zorlukların üstesinden gelmek de mümkündür. Uygulama noktasında analog sistemlerde oluşan problemlere çözüm olması amacıyla kaotik üreteçlerin sayısal donanım gerçekleştirmeleri kullanılabilir.

FPGA gibi programlanabilen lojik birimlerin hızlı ve kolay bir şekilde tekrar programlanabilme özellikleri modern sistemlerde bu donanımları, son derece popüler hale getirmiştir. Özellikle geçtiğimiz yirmi yıl boyunca, programlanabilir cihazlar sayısal sistemlerde yaygın olarak kullanılmış olup, sayısal devrelerin tasarlanmasında uzun zamandan beri önemli kolaylıklar sağlamaktadırlar. Özellikle veri güvenliği açısından sayısal bilgilerin şifrelenmesinde kaotik sistemlerden yararlanma fikri doğrultusunda yapılan çalışmalar referans gösterilerek, bu çalışmaların güvenli haberleşme sistemlerinde kullanılabileceği, geliştirilebileceği ve ayrıca FPGA tabanlı donanım kullanarak gerçekleştirilebileceği vurgulanmıştır.

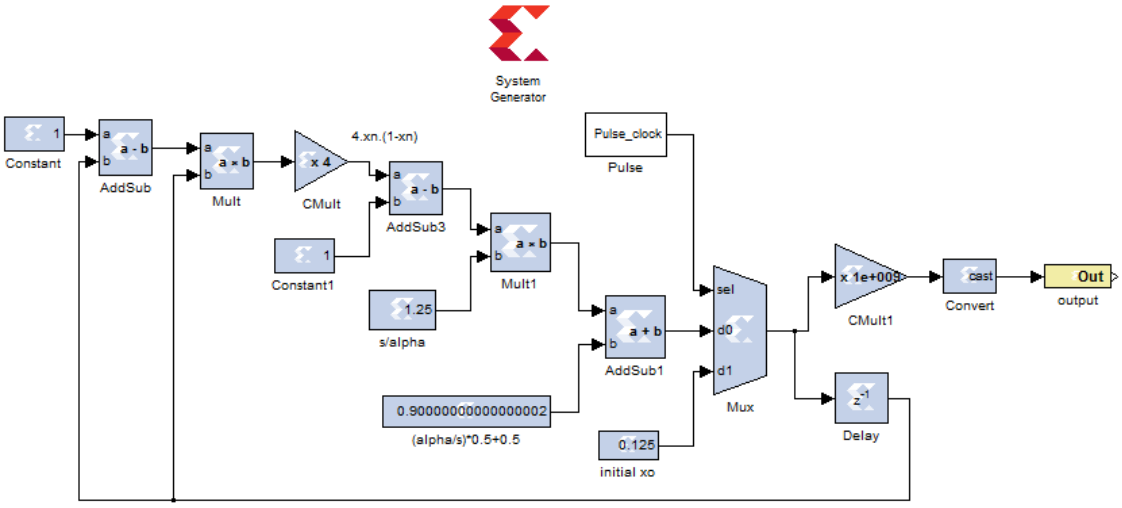
Bu kısımda, İLH sisteminin FPGA sisteminde gerçek zamanlı olarak çalıştırılmasına yönelik yaklaşım açıklanmaktadır. İLH sistemi, tasarlanan kripto sistemde kod üreteci olarak kullanılmaktaydı ve üreteç çıkışı her biri 8-bit uzunluğunda 0 ile 255 arasında değerlerden oluşmaktaydı. Kripto sistemin dağılma bölümünde belirtildiği gibi başlangıç değerleri ve kontrol parametreleri birbirinden farklı olan iki İLH üreteci şifreleme algoritmasında kullanılmıştır. Uygulama noktasında ise bunlardan birinin FPGA ortamında gerçek zamanlı çalıştırılması ve Simulink ortamında üretecin donanımsal benzetimi gerçekleştirilecektir.

Xilinx ISE yazılımı, MATLAB/Simulink programı ile bütünleşik çalışabilmektedir. Bu anlamda, İLH sistemini temsil eden matematiksel denklem, Simulink ortamında Xilinx blokları kullanılarak modellenmiştir. XSG aracı, Simulink ortamından FPGA birimine doğrudan uygulanabilen ve kullanıma hazır ondalıklı sayı aritmetiği kütüphanesine sahip Xilinx bloklar içerir. Xilinx kütüphanesinde bulunan bloklar Şekil 7.2’ de gösterilmiştir.



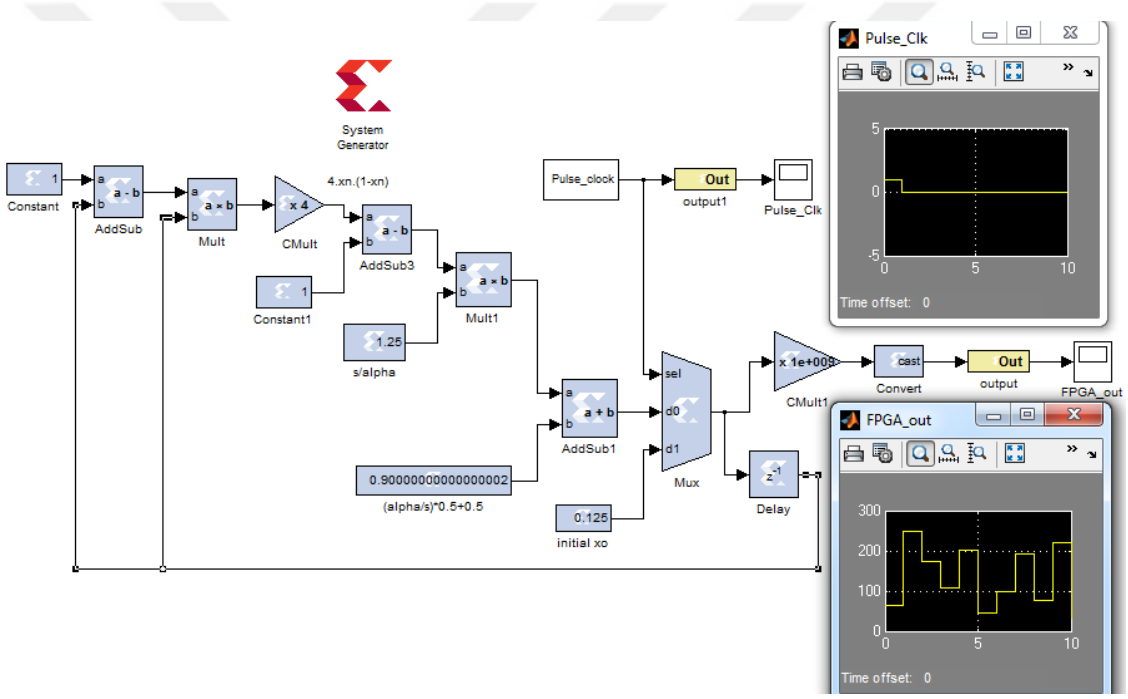
Şekil 7.2 Simulink ortamında bulunan Xilinx blokları

Xilinx blokları kullanılarak matematiksel tüm aritmetik işlemler yapılabildiği gibi aynı zamanda karmaşık fonksiyonlar da FPGA tasarımları için modellenebilir. Simulink ortamında Xilinx bloklarıyla oluşturulmuş İLH devre modeli Şekil 7.3' de verilmiştir. Burada İLH parametreleri: $s=100$, $\alpha=80$ ve $x_0=0.125$ gibi seçilmiş ve sistem çıktısı 0 ile 255 arasında olacak şekilde tasarlanmıştır.



Şekil 7.3 Xilinx blokları ile oluşturulmuş İLH sistemi

Modelde görüldüğü gibi bir tane çoklayıcı bloğu (Mux) kullanılmıştır. Çoklayıcı, en az iki girişten tek bir çıkış sağlayan bir fonksiyon gibi çalışır. Burada gerçekleştirilen yönlendirme işlemi ise çoklayıcı bloğuna bağlanan seçici bilgisine bağlıdır. İLH sistemi tekrarlayıcı bir denklem yapısına sahip olduğundan dolayı çıkış serilerinin üretimi için ilk olarak başlangıç değerinin girilmesi gerekir. Model içerisinde başlangıç değerinin seçilmesi, çoklayıcı bloğunun seçici girdisine bağlanan bir darbe sinyali ile sağlanmıştır. Yani darbe sinyali '1' iken başlangıç değeri olan d_1 girişi; '0' iken hesaplanan x_n değeri yani d_0 girişi, çıkış için seçilmelidir. Model için oluşturulan darbe sinyaline ve İLH üretecine ait Simulink çıkışı Şekil 7.4' de gösterilmiştir.



Şekil 7.4 Çoklayıcı bloğu için oluşturulmuş darbe sinyali ve İLH çıkışı

Daha önceden belirtildiği gibi İLH üretecinin çıkışında, Denklem (7.1)' de verilen ifade kullanılarak 8-bit uzunluğunda bir kod elde edilmekteydi.

$$kod = \mathrm{mod}(round(x_n \times 10^9), 256) \tag{7.1}$$

İLH çıkışına ilave edilen bu yapı, model ortamında çoklayıcı çıkışına sırasıyla sabit çarpım bloğu (CMult1) ve dönüştürücü bloğunun (Convert) eklenmesi ile sağlanmıştır. Burada

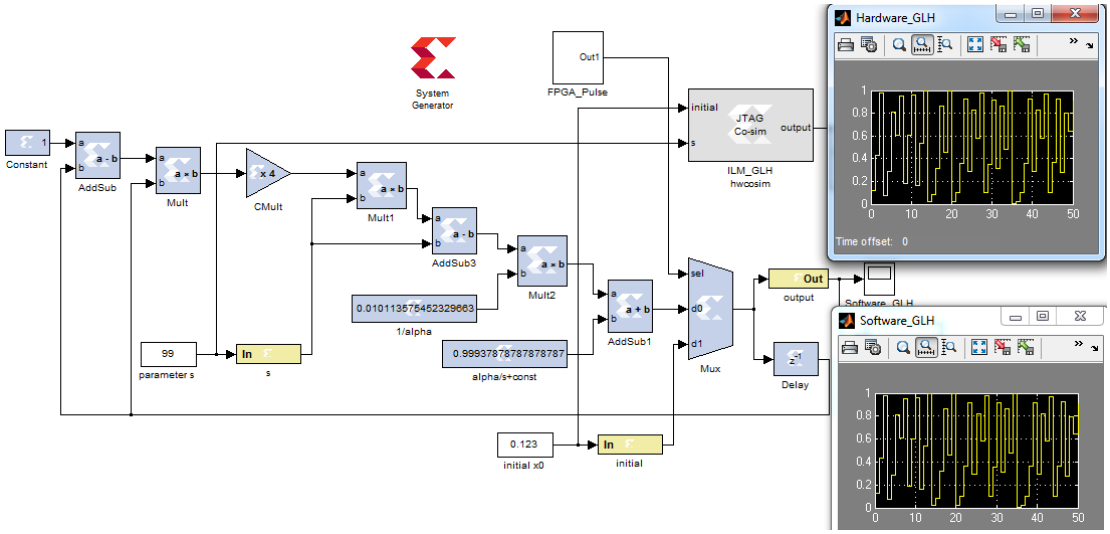
çarpım bloğu kendisine gelen sinyali 10^9 gibi bir değer ile çarpmakta ve dönüştürücü bloğu ise çarpım sonucu oluşan değeri yuvarlayıp, 8-bit değerine dönüştürmektedir. Çarpım sonucu 8-bit ile temsil edilemeyeceğinden dolayı bu işlem aynı zamanda modüler işleme (mod 256) denk gelmektedir. Sonuçta, Denklem (7.1)' de verilen 0-255 tamsayı kod dönüştürme için gerekli işlemler model ortamında da gerçekleştirilmiş olur. Örneğin, İLH üretecinin $x_0 = 0.125$ başlangıç değeri için Matlab ortamında hesaplanan ilk kod değeri Denklem (7.2)' den,

$$K_1 = \mathrm{mod}(round(0.125 \times 10^9), 256) \tag{7.2}$$

64 olmaktadır. Bu sonuç, Xilinx bloklarıyla oluşturulmuş Şekil 7.4' de verilen İLH modelindeki Simulink çıkışından da görülebilmektedir.

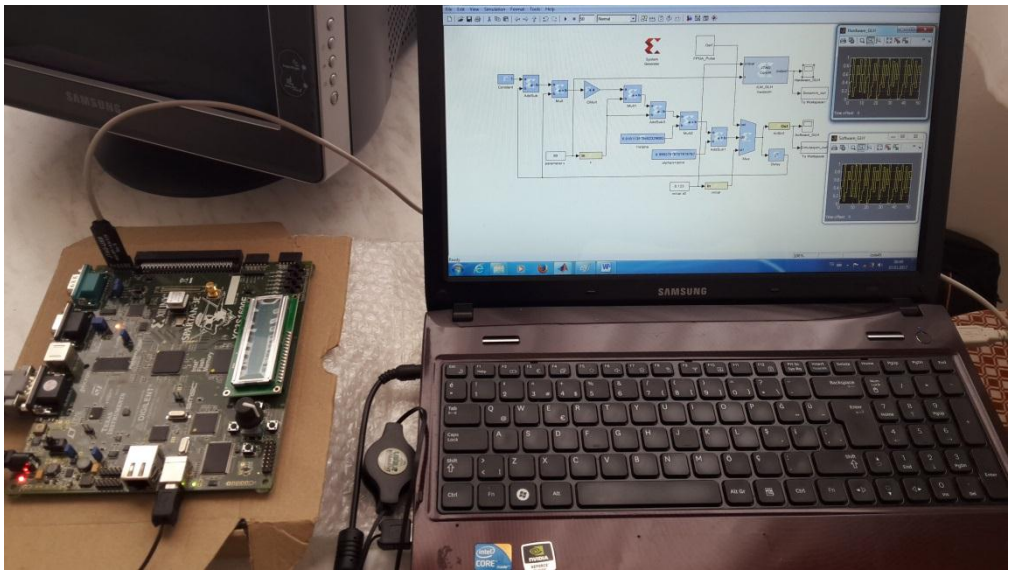
7.3.1 İLH Üretecinin Donanımsal Benzetimi

Donanımsal benzetim, Xilinx blokları ile oluşturulmuş model tabanlı tasarımın gerçekte FPGA platformunda çalıştığının kanıtı niteliğindedir. Bu kısımda, kripto sistemde kullanılan İLH sisteminin FPGA ortamındaki donanımsal benzetimi ele alınmıştır. Matlab yazılımı ile bütünleşik olarak çalışan XSG aracı kullanarak Simulink üzerinden ondalık tabanlı İLH sistemin donanımsal benzetim bloğu oluşturulmuş ve Simulink ortamında üretecin yazılımsal ve donanımsal benzetimi gerçekleştirilmiştir. Donanım tabanlı benzetim işleminin gerçekleşmesi için FPGA biriminin programlanması gerekir. Bu programlama işlemi benzetim başlangıcı sırasında XSG tarafından otomatik olarak gerçekleştirilir ve FPGA ile bilgisayar arasında JTAG kablosu üzerinden programlama dosyası aktarılır. İLH üretecine ait donanım bloğunu içeren FPGA modeli ve modele ait donanımsal benzetim sonuçları Şekil 7.5' de gösterilmiştir.



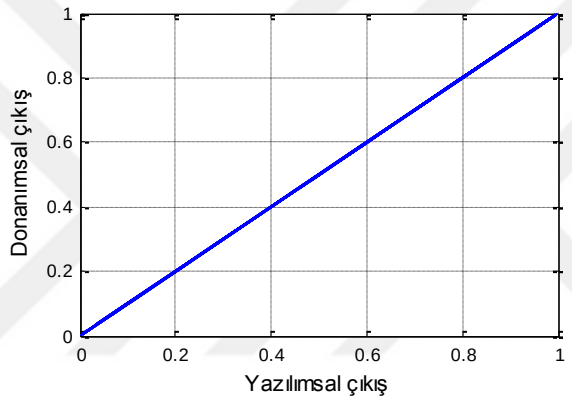
Şekil 7.5 İLH üretecinin donanımsal benzetimi

Uygulama sonunda, donanımsal ve yazılımsal benzetim sonuçlarının birbiriyle tamamen aynı olduğu gözlemlenmiştir. Bu sonuç, İLH üretecinin donanım tabanlı benzetiminin başarılı bir şekilde gerçekleştirildiğini ve üreteç tasarımının gerçekte FPGA donanımında çalıştığını göstermektedir. Deneysel çalışma sırasında çekilen bir görüntü Şekil 7.6' da paylaşılmıştır. Burada dikkat edilirse, kart üzerinde 'turuncu' ışığın yanması, FPGA sisteminin programlandığını göstermektedir.



Şekil 7.6 İLH üretecinin FPGA ortamındaki donanımsal benzetim uygulaması

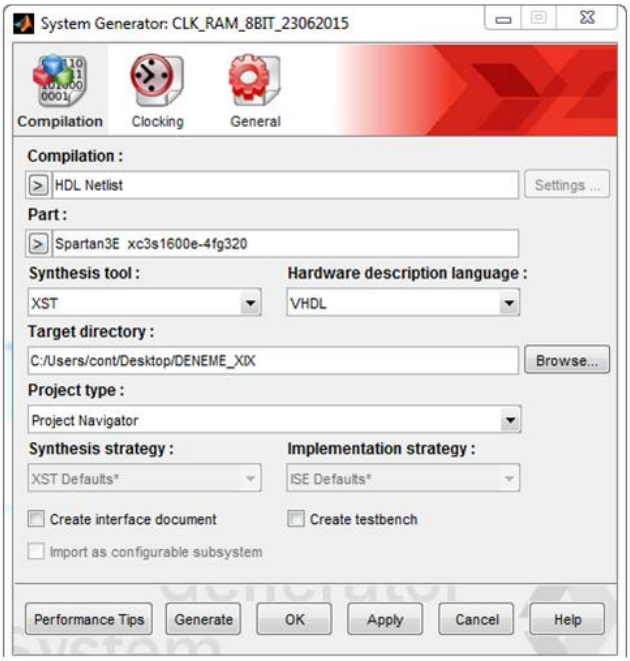
İLH üreteci ondalıklı işlemler gerektirdiğinden, bu değerlerin FPGA ortamında kaç bit ile temsil edildiği önemlidir. Burada donanımsal gerçekleştirmede her bir reel veri ve bloklara ait çıkış değerleri 118-bit ile temsil edilmiştir. Bu kısımda dikkat edilmesi gereken nokta, her bir verinin yüksek bit ile temsil edilmesi halinde donanımdaki kaynak tüketimini arttıracağı; az sayıda bit ile temsil edilmesi durumunda ise doğruluk noktasında donanımsal benzetim sonuçlarında farklılık oluşturabileceğidir. Bu anlamda, veri temsili ve işlemlerinde optimum olarak 118-bit tercih edilmiştir. Şekil 7.7' de verilen grafik, 1000 iterasyon için donanım (FPGA) ve yazılım (MATLAB) benzetim sonuçlarını karşılıklı olarak göstermektedir.



Şekil 7.7 İLH çıkışının FPGA ve MATLAB ortamında karşılaştırılması

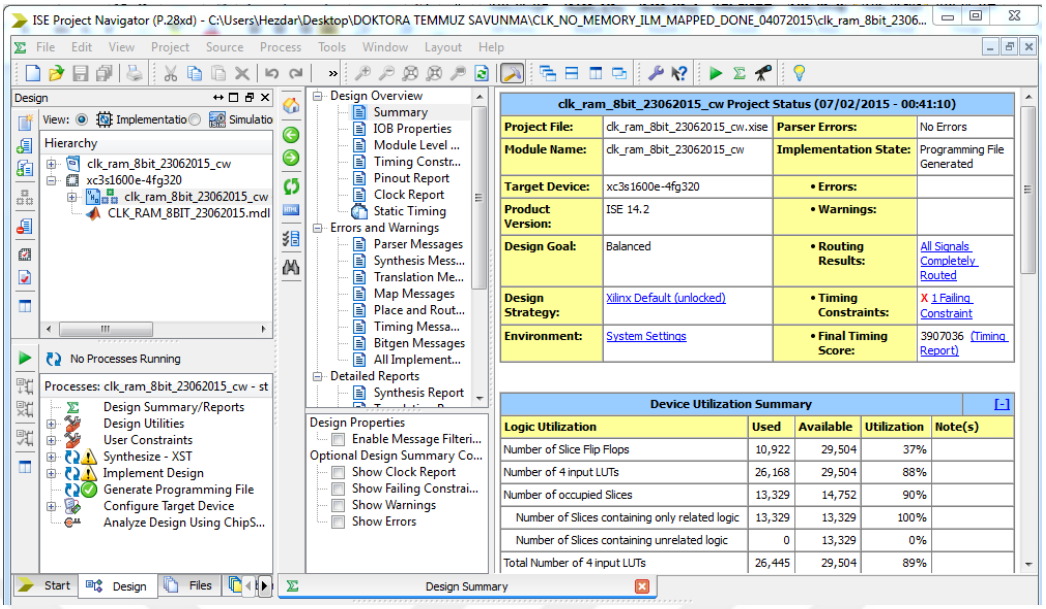
7.3.2 İLH Üretecinin Gerçek Zamanlı FPGA Uygulaması

Bu bölümde, İLH kod üretecinin gerçek zamanlı olarak FPGA donanımında çalışması ele alınacaktır. Gerçek zamanlı uygulama için Şekil 7.4' de oluşturulan İLH modeli kullanılmıştır. İLH modelinde XSG üzerinden donanım tanımlama dili (HDL) derlemesi ilk olarak yapılmıştır. Bu işlem sırasında XSG, tasarımla ilgili olarak Xilinx ISE yazılımda açılabilen ve sentezlenebilen VHDL kodlarını otomatik oluşturur. Yapılan bu işlemlerle ilgili ekran görüntüleri Şekil 7.8' de verilmiştir.



Şekil 7.8 İLH modelinin XSG üzerinden derlenmesi

XSG, devre tasarımlarıyla ilgili olarak menü üzerinden derleme seçeneğini, FPGA modelini, kullanılacak donanım tanımlama dilini ve işlem sonunda gerekli dosyaların nerede oluşturulacağı ile ilgili seçenekler sunmaktadır. Uygulamada kullandığımız FPGA sınıfı ve donanım dili ekran menüsünden de görüldüğü gibi seçilmiştir. Şekil 7.9' da ise derleme sonucu oluşan VHDL dosyasının, ISE yazılımında açılması ile ilgili görüntü verilmiştir. Burada dikkat edilirse, ISE içerisinde çalışma dosyasıyla ilgili Simulink modeline ulaşılabileceği de görülebilmektedir.



Şekil 7.9 VHDL dosyasının ISE yazılımıyla açılması

Sentezleme, benzetim ve programlama dosyasını oluşturma gibi tasarıma yönelik önemli işlemler ISE yazılımda gerçekleştirilebilir. Ayrıca ISE yazılımı içinde bulunan 'PlanAhead' aracıyla FPGA sisteminin programlanmasına yönelik olarak tasarıma ait giriş ve çıkış yerleri atanabilir. Örneğin, İLH anahtar üretecimizin çıkışı 8-bit uzunluğundadır. Bu çıkış, FPGA donanımın LED çıkışına bağlı olan pin adreslerine doğru olarak atandığında her bir iterasyonda İLH sisteminin ürettiği kod, gerçek zamanlı olarak FPGA donanımının LED çıkışında ikili formda gözlemlenebilir.

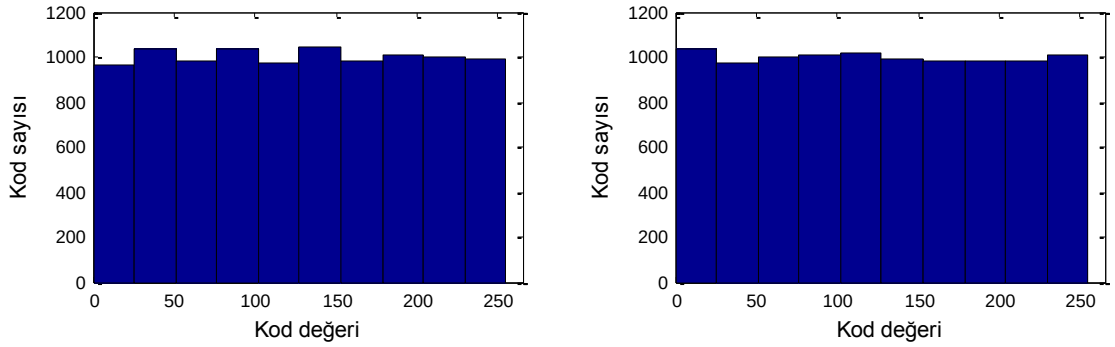
İLH üretecisi için donanımsal benzetimde kullanılan sistem parametreleri, $s=100$, $\alpha=80$ ve $x_0=0.125$ gerçek zamanlı FPGA uygulamasında da aynen kullanılsın. Burada MATLAB ve FPGA ortamında, İLH üretecinden ilk 35 iterasyon için elde edilen kod değerleri Tablo 7.1' de listelenmiştir.

Tablo 7.1 MATLAB ve FPGA ortamında üretilen şifreleme kodları

n	**MATLAB**	**FPGA**
1	64	64
2	248	248
3	172	172
4	107	107
5	203	203
6	44	44
7	98	98
8	191	191
9	76	76
10	219	219
11	30	30
12	81	81
13	169	169
14	38	38
15	153	153
16	34	34
17	39	39
18	204	204
19	39	39
20	48	48
21	143	143
22	164	164
23	30	30
24	111	111
25	133	133
26	76	76
27	**147**	**145**
28	**162**	**157**
29	**175**	**176**
30	**187**	**182**
31	**57**	**35**
32	**58**	**239**
33	**93**	**189**
34	**72**	**91**
35	**218**	**36**

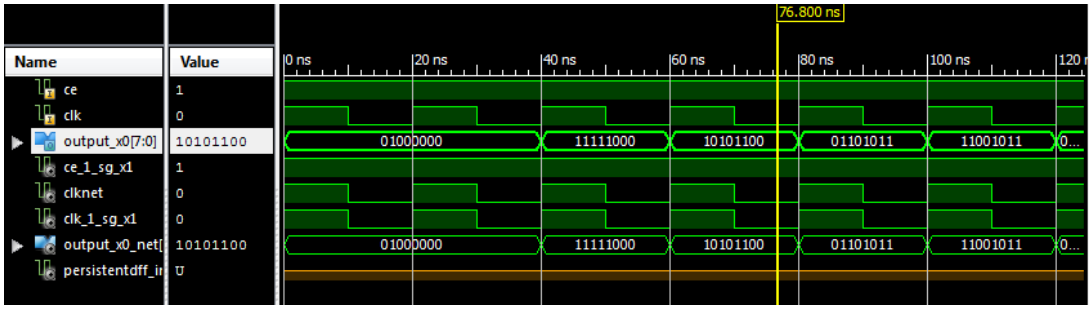
Tablo 7.1' de görüldüğü gibi aynı parametre değerleri ile MATLAB ve FPGA ortamında çalıştırılan İLH üreteci, 27. iterasyondan sonra farklı platformlarda tamamen farklı kod değerleri üretmektedir. IEEE ondalık sayı aritmetiği standardına (754) göre MATLAB, bilgisayar sayı formatı olarak geniş aralıklı ondalık işlemlerde 64-bit kullanır. Bu çalışmada ise İLH üretecinin FPGA tasarımında, Xilinx bloklarıyla gerçekleştirilen ondalık işlemlerin 118-bit ile temsil edildiği belirtilmişti. Bu farktan dolayı her bir iterasyonda artan hata, 10^{-9} değerinden büyük olduğu takdirde, İLH çıkışında tamsayı dönüşümde kullanılan '$\mathrm{mod}(round(x_n \times 10^9), 256)$' matematiksel işlemi farklı sonuç verecektir. Bu değişim, verilen tablodan fark edildiği gibi 27. iterasyondan sonra kendini göstermiştir.

Şekil 7.10’ da gösterilen histogram grafiği ise MATLAB ve FPGA ortamlarında 10,000 iterasyon için elde edilen kod değerlerinin dağılımını vermektedir.



Şekil 7.10 Farklı platformlardan üretilen kodların histogram dağılımları; (a) MATLAB, (b) FPGA

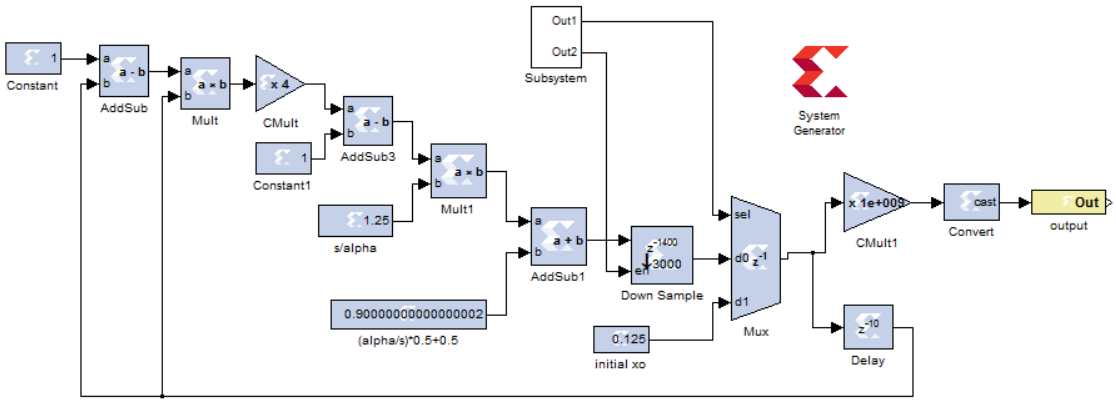
Grafiksel sonuçlardan da kolayca görülebildiği gibi aynı sistem parametreleri ile iki farklı ortamda üretilen kodlar benzer karakteristik özellikte dağılmışlardır. İLH üretecinin gerçek zamanlı FPGA uygulamasında ise öncelikli olarak ISE ortamında açılmış İLH modeliyle ilişkilendirilmiş VHDL kodunun başarılı bir şekilde sentezlenmesi ve programlama dosyasının oluşturulması gerekir. Bu noktada eğer sentezlenen devre modelinin kaynak ihtiyacı, FPGA donanımının sahip olduğu sınırlı kaynaktan fazla ise sentezleme başarısız olur. Simulink ortamında Xilinx kütüphanesinin, ‘*Resource Estimator*’ bloğu, modellenen devrenin kaynak ihtiyacını tahmini olarak belirleyebilir. Başarılı sentezlemeden sonra programlama dosyası otomatik olarak oluşturulur. Bit uzantılı bu dosya, ISE yazılımının IMPACT aracı üzerinden hedeflenen FPGA sistemine gönderilerek donanım programlanır. Ayrıca ISIM üzerinden tasarlanan modelin davranışsal benzetimi de gerçekleştirilebilir. Böylelikle devre modelinin çalışma akışı donanıma aktarılmadan önce benzetim ortamında gözlemlenebilir. İLH üretecinin ISIM üzerinde gerçekleştirilen benzetim sonucu Şekil 7.11’ de gösterilmiştir.



Şekil 7.11 İLH üretecinin ISIM üzerindeki benzetimi

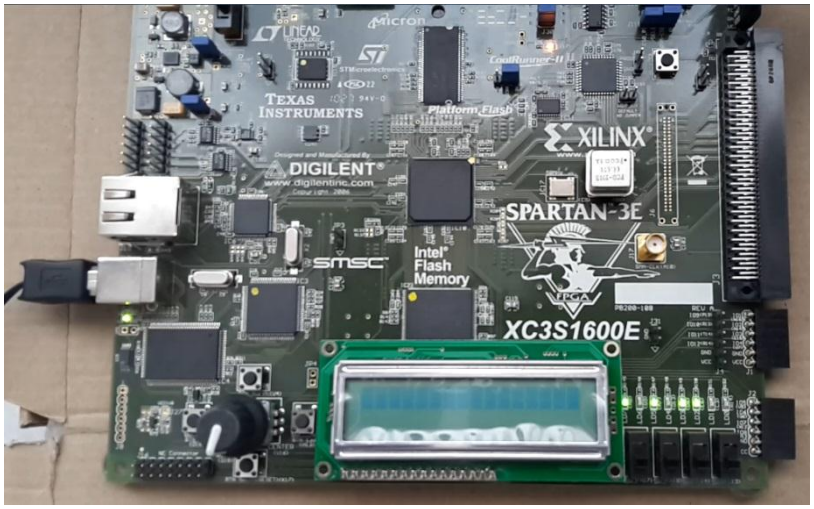
İLH devre modelinin yazılım ortamındaki benzetim sonucundan görülebildiği gibi üreteçten birinci, ikinci ve üçüncü iterasyonda elde edilen kod değerleri sırasıyla '01000000', '11111000' ve '10101100' dir. İkili düzendeki bu değerler dikkat edilirse, MATLAB ve FPGA ortamında üretilen ve Tablo 7.1' de gösterilen 64, 248 ve 172 değerlerine denk gelmektedir. Bu sonuç, İLH modelinin benzetim davranışının başarıyla gerçekleştirildiğini ve FPGA donanımının, modele ait programlama dosyası ile programlandığında benzetimdeki değerlerin FPGA çıkışında da gözlemleneceğini açıklamaktadır.

Benzetim grafiğinden ayrıca her 20 ns' de, yeni bir kod değerinin üretildiği görülebilmektedir. Kullanılan FPGA sınıfının 'C9' pin adresi mevcut 50 MHz hızında bir sinyal üretir. Bu frekans değeri ise 20 ns olan bir zaman periyoduna denk gelmektedir. Yani FPGA sistemi, her 20 ns' de yeni bir işlem gerçekleştirerek bir sonraki kod değerini hesaplamaktadır. Sonuç olarak, başarılı bir sentezlemeden sonra programlama dosyasının hedef birimi yapılandırılmasıyla İLH üreteci gerçek zamanlı olarak FPGA ortamında çalışmaktadır. Gözlem kolaylığı sağlamak adına her bir kodun üretilme zamanını 20 ns' den yeterince büyük bir seviyeye çıkartılırsa, o zaman üreteç çıktısı 8-bit olarak LED çıkışında kolayca gözlemlenebilir. Bu amaca yönelik olarak İLH üretecinin Simulink modeline, sistemin örnekleme oranını düşüren '*Down Sample*' bloğu, Şekil 7.12' de gösterildiği gibi kullanılmıştır.



Şekil 7.12 Uygulama noktasında kullanılan İLH devre modeli

Örneğin, İLH üreteci verilen parametreler altında üçüncü iterasyonda, Tablo 7.1' de verildiği üzere 172 kod değerini üretmiştir. Gerçek zamanlı uygulamaya yönelik olarak bu değerin ikili sistemdeki karşılığı olan '10101100' değerinin FPGA ortamında LED çıkışında gözlemlenmesine ilişkin uygulama görüntüsü Şekil 7.13' de verilmiştir. İLH modelinin sentezleme sonrası FPGA donanımının kaynak kullanımına ait rapor ise Tablo 7.2' de verilmiştir.



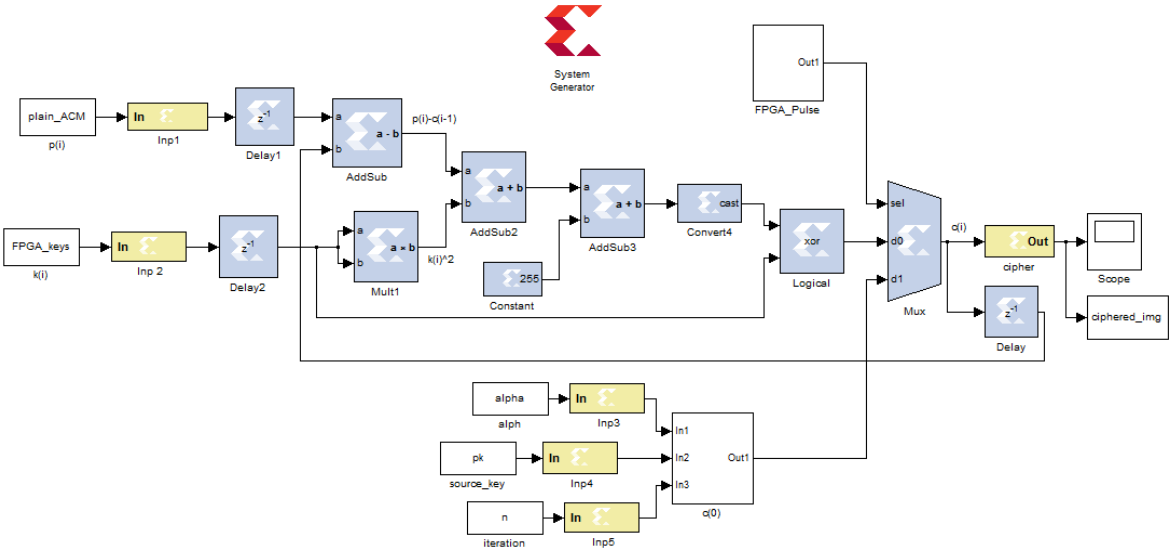
Şekil 7.13 FPGA donanımına ait LED çıkışında 172 değerinin ikili tabanda gösterimi

Tablo 7.2 İLH kod üretecinin kaynak kullanımı raporu

Donanım	**Spartan 3E-XC3S1600E**						
Kaynak Sınıfı	***Slices***	***Flip-Flops***	***RAMB 16S***	***LUTs***	***IOBS***	Çalışma Frekansı (Mhz)	Bellek Kullanımı (MB)
Mevcut	14,752	29,504	36	29,504	250	18.7	690
Kullanılan	13,329	10,922	1	26,445	9		
Yüzdesi (%)	90	37	2	88	3		

7.4. Şifreleme ve Çözme Algoritmalarının FPGA Ortamında Gerçekleştirilmesi

Bu kısımda, tasarlanan görüntü kripto sistemine ait şifreleme ve çözme algoritmalarının FPGA ortamında gerçekleştirilmesine yönelik işlemler ele alınacaktır. Öncelikle kripto sistemde kullanılan şifreleme algoritması, Simulink ortamında Xilinx blokları kullanarak modellenmiştir. Kripto sistemin dağılma bölümünde anlatıldığı gibi piksellerin değiştirilmesinde Denklem (5.6)' da verilen bir şifreleme fonksiyonu kullanılmıştı. Buradaki matematiksel ifadeden görüldüğü gibi şifreleme fonksiyonu XOR ve modüler işlemlerin dışında basit aritmetik işlemlerden de oluşmaktadır. Bu işlemlerin tamamı Simulink ortamında Xilinx blok setleri ile tanımlanabilir. Deneysel uygulamalarda pratiklik sağlaması amacıyla FPGA ortamında gerçekleştirmelerde şifreleme ve çözme algoritmaları için $k_1(i) = k_2(i)$ durumu benimsenmiştir. Yani her iki işlemde de bir İLH kod üreticisi kullanılmış ve tek iterasyon uygulanmıştır. Burada hedeflenen amaç sunulan şifreleme ve çözme algoritmalarının donanım ortamında başarıyla sentezlenmesini sağlamak ve uygulanabilirliliğini denetlemektir. Dolayısıyla bu amaç sağlandığı takdirde benzer tüm kriptografik sistemlerin FPGA ortamında pratik uygulamalara yönelik kullanılabilirliliği doğrulanmış olacaktır. Kripto sistemin şifreleme algoritmasını temsil eden dağılma fonksiyonunun Simulink ortamında Xilinx blokları kullanarak oluşturulmuş modeli Şekil 7.14' de gösterilmiştir.

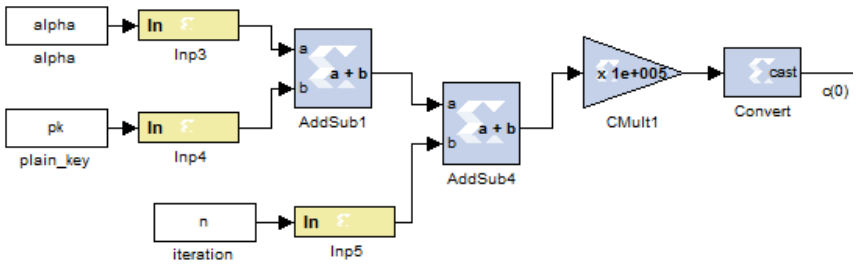


Şekil 7.14 Şifreleme algoritmasının Xilinx bloklarıyla modellenmesi

Şekil 7.14' de verilen devre modelinin giriş bilgileri: FPGA donanımından üretilen şifreleme kodları ve kaynak görüntünün AKH çıkışıdır. Ayrıca bu girişlerin hemen ardından birer gecikme bloklarının kullanıldığı görülebilmektedir. Şifreleme senkronizasyonunun sağlanması için bu gereklidir. Çünkü algoritmanın tekrarlayıcı bir yapıya sahip olması, ilk şifreli piksel değerini gecikmeli olarak şifreleme işlemine sokmaktadır. Bunun dışında, şifreleme işleminin başlatılabilmesi için ilk şifreli piksel değeri olarak tanımlanan $c(0)$ değerinin de model ortamında oluşturulması gerekir. Bu değerin matematiksel karşılığı Denklem (5.7)' de tanımlanmıştı. Burada örnek uygulamada, tek bir İLH üreteci kullanıldığından, şifreleme algoritması için bu parametre değeri Denklem (7.3)' deki gibi olacaktır.

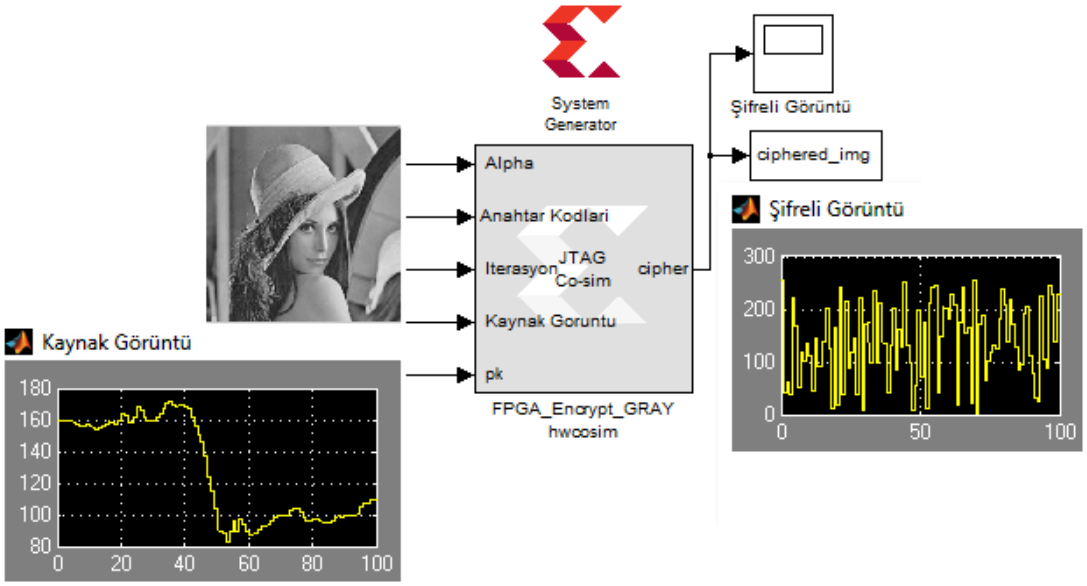
$$c(0) = \mathrm{mod}(round(\alpha + pk + n) \times 10^5), 256) \tag{7.3}$$

Bir şifreleme kodu olarak da düşünülebilen $c(0)$ parametresi, kaynak görüntü ile ilişkili pk parametresi dışında kripto sistemin s, x_0 ve n şifreleme anahtarlarına bağlıdır. Dolayısıyla $c(0)$ parametresinin giriş değerleri, MATLAB ortamından gerekli anahtarların çağrılmasıyla sağlanabilir. Buna göre bu parametre değerinin Xilinx bloklarıyla oluşturulmuş Simulink modeli Şekil 7.15' de gösterilmiştir.



Şekil 7.15 Şifreli ilk değerin Xilinx bloklarıyla oluşturulması

Şekil 7.14 ve Şekil 7.15' de verilen modeller şifreleme algoritmasını tamamlamaktadır. Algoritmanın belirli bir kripto sistem parametreleriyle çalıştırılması ve bu işlemin FPGA ortamında gerçekleştirilmesi için şifreleme işleminin donanımsal benzetiminden yararlanılmıştır. Yani şifreleme işleminin tamamı FPGA biriminde gerçekleşmekte ve bu işleme ait sonuçlar ise Simulink ortamında hızlı bir şekilde gözlemlenebilmektedir. Ayrıca, donanım tabanlı benzetim ile birlikte FPGA biriminde işlenen çıkış verileri, Simulink ortamında saklanması veya uygun veri formatına çevrilmesi gibi farklı amaçlara yönelik olarak da kullanılabilir. Donanımsal benzetim bloğu oluşturulmuş şifreleme yapısının Simulink ortamındaki benzetim sonucu Şekil 7.16' da verilmiştir. Bu uygulama çalışmasında test görüntüsü olarak 256×256 ölçüsündeki 'Lena' görüntüsü kullanılmıştır. Deneysel uygulama için kaynak görüntüde tam bir şifreleme işlemi gerçekleştirilmiş olup, sadece ilk 100 iterasyon için grafiksel sonuç paylaşılmıştır.

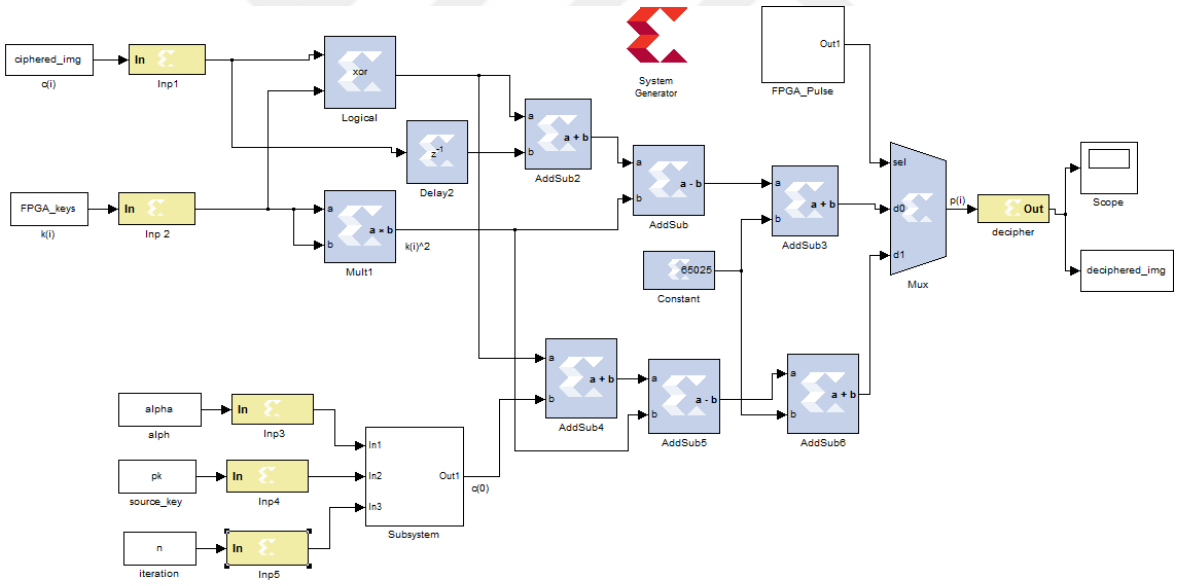


Şekil 7.16 'Lena' görüntüsü için şifreleme algoritmasının donanımsal benzetimi

Şifreleme algoritmasına ait donanım bloğunun giriş parametrelerinden *alpha*, *pk* ve *n* değerleri şifreleme işlemini başlatacak olan $c(0)$ ilk şifreli piksel değerini oluşturmak için kullanılmıştır. Şifrelemede kullanılan anahtar kodları ise FPGA donanımından üretilmiştir. Tasarlanan kripto sistem simetrik bir algoritmaya sahip olduğundan dolayı çözücü algoritmada aynı şifreleme kodlarının ve $c(0)$ değerinin kullanılması gerekir. Çalışmanın uygulama tarafında tek kod üreteci kullanıldığından dolayı kripto sistemin çözme algoritması, Denklem (7.4)' deki gibi olacaktır.

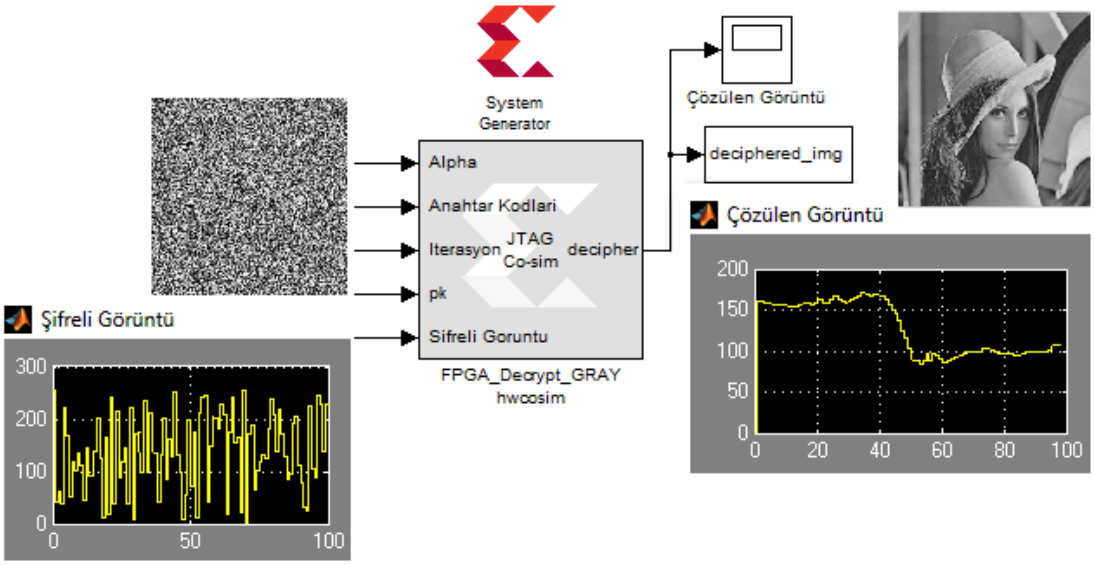
$$p(i) = \left\{ (k(i) \oplus c(i)) + c(i-1) - k(i)^2 + 255^2 \right\} \bmod 256 \tag{7.4}$$

Şifrelenmiş görüntü üzerinde çözme işlemi gerçekleştiren ve Xilinx blokları ile tasarlanmış devre modeli Şekil 7.17' de gösterilmiştir.



Şekil 7.17 Çözme algoritmasının Xilinx bloklarıyla modellenmesi

Uygulama sonucu Şekil 7.16' da gösterilen deneysel örnekte 'Lena' test görüntüsünün şifrelenmesiyle ilgili FPGA gerçekleştirilmesi yapılmıştı. Benzer şekilde şifrelenmiş 'Lena' görüntüsü üzerine çözme algoritmasıyla ilişkilendirilmiş donanımsal benzetim uygulanmış ve sonuçlar Şekil 7.18' deki gibi gözlemlenmiştir.



Şekil 7.18 Şifrelenmiş 'Lena' görüntüsü için çözme algoritmasının donanımsal benzetimi

Deneysel uygulama sonuçlarına göre sunulan kripto sisteme ait şifreleme ve çözme algoritmaları başarılı bir şekilde FPGA ortamında gerçekleştirilmiş olup, şifrelenmiş 'Lena' görüntüsü, çözme algoritmasında aynı şifreleme anahtarları kullanılarak doğru bir şekilde tekrar elde edilmiştir.

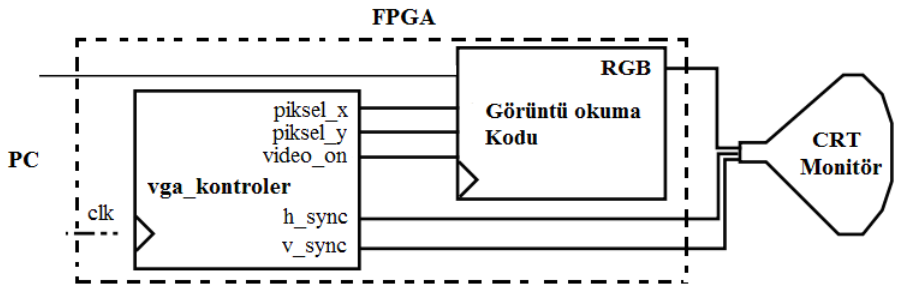
7.5. FPGA Üzerinde VGA Sinyal Üretimi ve Görüntü Aktarımı

Bu çalışmada, anahtar üreteci olarak tasarlanan İLH üretecinin modellenmesinde sadece Xilinx kütüphanesine ait bloklar kullanılmıştır. Benimsenen donanım dili için standartlaştırılmış bir ondalık sayı kütüphanesinin olmaması, kod tabanlı VHDL yazılımını fazlasıyla güçleştirmektedir. Bu açıdan değerlendirildiğinde, Simulink ortamında devre modellemelerinde büyük esneklik ve kolaylık sunmasıyla birlikte sistem davranışını donanımsal benzetime olanak sağlaması, İLH üretecinin XSG blokları kullanarak modellenmesinde büyük avantaj sağlamaktadır. Sunulan kripto sistemin kısımları olarak anahtar üreticisi, şifreleme ve çözme algoritmaları VHDL dili kullanmadan sadece Xilinx blokları ile Simulink ortamında modellenmiştir. Çalışmanın uygulama kısmının ikinci aşamasında ise FPGA ortamında gerçekleştirilen şifreleme ve çözme işlemleri sonucunda oluşan görüntülerin VGA portu üzerinden dış birim olan bir monitöre aktarılması amaçlanmıştır. Burada görüntülerin aktarılmasında VGA görüntü standardı kullanılmıştır.

VGA, çoğu modern bilgisayar ekranında ortak olan bir video görüntü standardıdır. Bu arayüz, herhangi bir bilgi veya görüntüyü bir sistem üzerinden monitöre bağlamak için

bir yöntem sağlar. Bir VGA monitör kırmızı, yeşil, mavi, yatay senkronizasyon ve dikey senkronizasyon olmak üzere beş sinyal tarafından kontrol edilir. Topluca RGB sinyali olarak adlandırılan bu üç renk sinyali, ekrandaki belirli bir konumdaki pikselin rengini kontrol eder. Bunlar, gerilim değerleri 0.7 ile 1 volt arasında değişen analog sinyallerdir. Yatay ve dikey senkronizasyon sinyalleri ise sayısal olup, ekran tarama hızının zamanlamasını kontrol etmek için kullanılır. Daha açık bir ifadeyle, yatay senkronizasyon sinyali, bir satırı taramak için geçen süreyi belirlerken; dikey senkronizasyon sinyali ise tüm ekranın taranması için geçen süreyi belirler. Bu iki senkronizasyon sinyali ve üç RGB sinyali değiştirilerek görüntüler monitör ekranında oluşturulur [109].

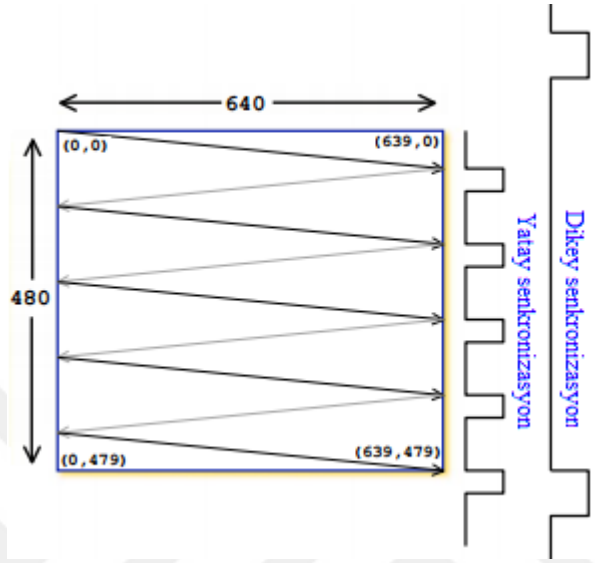
Kaynak ve şifrelenmiş görüntünün, FPGA donanımının VGA portu üzerinden harici bir monitöre aktarılmasında öncelikle yazılımsal bir VGA kontrolör gereklidir. VGA kontrolör, VGA sinyal zamanlamasını yöneten, görüntü ile ilgili olarak çözünürlük ve yenileme oranını yapılandıran bir yazılımdır. Bu çalışmada, görüntünün aktarılmasında tasarlanan VGA kontrolör, VHDL dili ile yazılmıştır. Oluşturulan VGA kontrolör ile birlikte çalışmada kullanılan tüm dosya ve programlar tez kitapçığında ek olarak CD içerisinde verilmiştir. Şekil 7.19' da, FPGA donanımının VGA portu üzerinden çıkış birimi olarak CRT (Cathode Ray Tube) analog monitöre görüntü aktarımının genel blok şeması gösterilmiştir.



Şekil 7.19 FPGA üzerinden görüntü aktarımının genel şeması

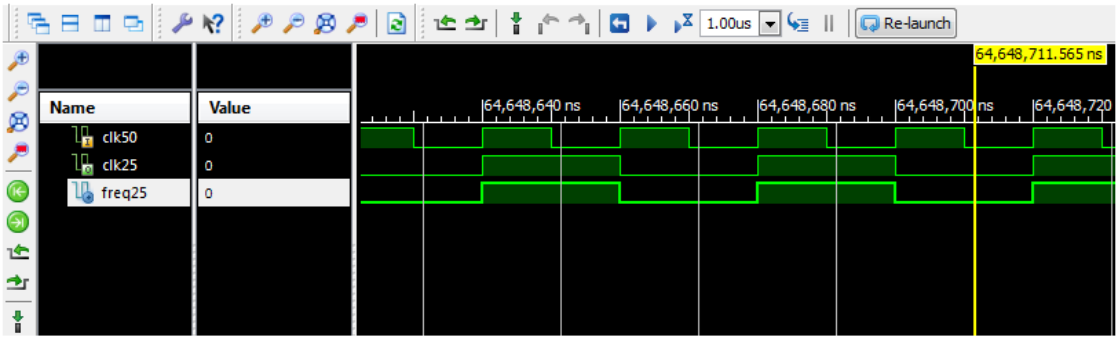
Monitör üzerinde herhangi bir görüntü, her bir pikselinin açık veya kapalı olmasıyla ekranda belirir ve monitör sürekli olarak ekranın tamamını çok hızlı bir şekilde tarar. Video sinyalleri, görüntü hareketinin sağlanması ve ekrandaki titremenin azaltılması için ekranın tamamının saniyede en az 60 kez taranması gerekir ki, bu periyot yenileme oranı olarak adlandırılmaktadır. Ekranda taranma işlemi, ekranın sol üst köşesindeki 0 satır 0 sütunundan başlar ve son sütuna ulaşana kadar sağa doğru gider. Ekranın sağ alt

köşesindeki son piksele ulaştığında tekrar sol üst köşeye geri döner ve tarama işlemi tekrarlanır. Böyle bir tam tarama saniyede 60 kez gerçekleşir. Bu işlemin gerçekleşmesini anlatan şema Şekil 7.20' de gösterilmiştir.



Şekil 7.20 VGA kontrolörde ekranın taranma işlemi

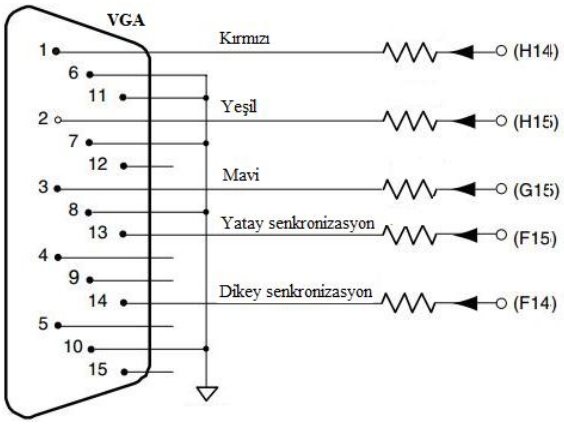
Burada aynı satırda tarama sırasında yatay senkronizasyon sinyali lojik olarak '0' iken, lojik '1' durumuna geçtiğinde tarama bir alt satıra geçmektedir. Benzer şekilde ekran tam olarak bir kez tarandığında yani konum olarak (639,479) pozisyonunda iken dikey senkronizasyon sinyali lojik '1' durumuna geçer ve ekran en baştan yeniden taranmaya başlar. Örneğin, 640×480 piksel çözünürlükte, 60 Hz yenileme oranı piksel başına 40 ns süre gerektirir, bu da 25 Mhz frekansına denk gelmektedir. Dolayısıyla öncelikli olarak 25 Mhz seviyesindeki frekans değerinin FPGA ortamında üretilmesi gerekir. Bunun için kullanılan FPGA kartı üzerinde standart olarak bulunan 'C9' pin adresinin sağladığı 50 Mhz ana frekansdan yararlanılmıştır. Ancak bu frekans değeri, gerekli olan piksel frekansının iki katıdır. Dolayısıyla ana frekansın periyodu iki katına çıkartılırsa ihtiyacımız olan 25 Mhz frekans elde adilmiş olur. Bunun için VHDL ortamında uygun frekans bölücü bir kod yazılmıştır ve bu programın sentezlenip, ISIM ortamında çalıştırılmasıyla elde edilmiş benzetim sonucu Şekil 7.21' de verilmiştir.



Şekil 7.21 25 Mhz frekansına ait ISIM benzetim sonucu

Benzetim sonucundan da görülebildiği gibi seçili '*freq25'* çıkış sinyalinin periyodu 40 ns olup, frekans seviyesi ana frekans ('*clk50*') değerinin yarısıdır. Kısaca değeri 25 Mhz olan bir sinyal, FPGA ortamında üretilmiş durumdadır. Böylelikle her bir pikselin ekranda kalması gereken süre bu sinyalin periyodu ile sağlanmış olacaktır.

VGA standardında her renk pikseli 3-bit ile temsil edilir. Uygulamalarda kullandığımız Spartan 3E-XC3S1600E sınıfı, VGA çıkış sağlayabilen bir FPGA' dır. Dolayısıyla görüntüdeki her bir piksel 8 farklı renk ile gösterilebilir. FPGA biriminin VGA portu için tanımlı adresler ve 3-bit RGB sinyalinin karşılığı olan renk kodları Şekil 7.22' de verilmiştir.



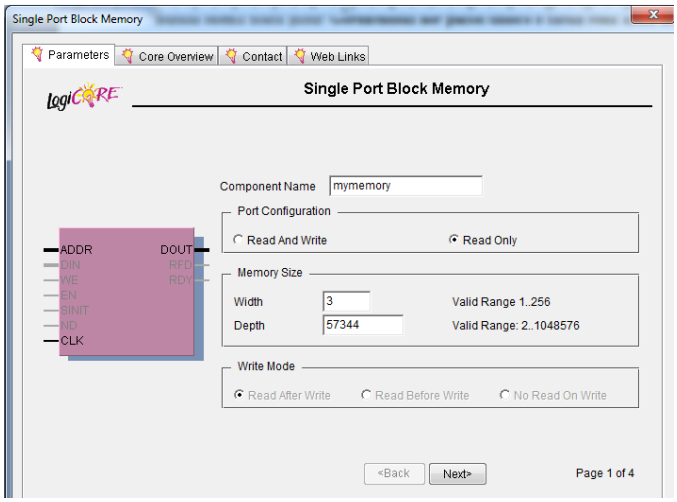
KIRMIZI	YEŞİL	SARI	RENK
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Şekil 7.22 VGA bağlantı şeması ve RGB renk kodları

Burada görüldüğü gibi VGA portuna ait H14, H15, G15, F15 ve F14 pin adresleri sırasıyla Kırmızı, Yeşil, Mavi renk bilgileri ile yatay ve dikey senkronizasyon sinyallerinin girişleridir. RGB sinyali, ilgili pin adreslerine '1' veya '0' sürülerek oluşan değere karşılık gelen renk pikseli üretilir. Dolayısıyla görüntüdeki her piksel sadece sekiz farklı renk kodu

ile temsil edilebilir. Matlab ortamındaki teorik çalışmalarımızda ise kaynak görüntü olarak kullanılan 8-bit seviyesinde gri görüntülerde her bir renk pikseli 256 farklı renk kodu ile temsil edilmekteydi. Dolayısıyla bu görüntülerin VGA portu üzerinden aktarılmasında kalite kaybının gerçekleşeceği açıktır. Başka bir ifadeyle kaynak görüntüdeki 8-bit renk haritası, VGA standardı için 3-bit seviyesine haritalanması gerekir. Bu dönüştürme ile ilgili işlem ilerleyen kısımda detaylandırılacaktır.

VGA kontrolör için yazılmış VHDL koduna bakıldığında (bkz. CD), *clk_25*: ana clock frekansını yarıya düşüren sinyali; *hsync* ve *vsync* zamanlama ve senkronizasyon sinyallerini; *srow* ve *scol* taramanın yatay ve dikey pozisyonunu yani ilgili pikselin mevcut konumunu; *pon* lojik sinyali ise görüntünün oluşacağı ekranın aktif olup olmadığını gösterir. Burada *pon* sinyali lojik '1' olduğunda ekran görüntü için hazırdır ve her bir pikseli temsil eden 3-bit uzunluğundaki RGB verisi VGA portuna gönderilmesi gerekir. Aynı sinyal lojik '0' olduğunda ise ekran pasif durumdadır. Bu uygulamada RGB verilerinin saklanacağı ve okunacağı birim, FPGA kaynaklarından 'Tek Port Blok Bellek' (Single Port Block Memory) yapısıdır. Bu modül, kullanıcı tarafından belirlenir ve görüntü ile ilgili genişlik ve derinlik bilgisi temel alınarak oluşturulur. Yani buraya görüntü bilgisinin yanında verinin saklanacağı bellek ölçüsü ile ilgili bilgiler girilmektedir. Örneğin, gerçekleştirilen bir uygulamada 256×224 ölçüsüne sahip bir test görüntüsü kullanılmıştır. Böyle bir görüntüde toplam 57344 piksel bulunur ve her bir piksel, 3-bit uzunluğundadır. Bu değerlerin modüle girilmesiyle ilgili ait arayüz Şekil 7.23' de gösterilmiştir.



Şekil 7.23 Görüntü bilgisinin saklanacağı tek port blok bellek kullanımı

İyi bir VGA kontrolör, görüntünün CRT monitöre aktarılmasında tek başına yetmez. Bunun yanında görüntüyü okuyan bir kod yapısına ve görüntü bilgisinin saklanacağı bir alana ihtiyaç vardır. Deneysel çalışmalarda görüntü bilgisinin depolanacağı alan için ISE yazılımın "Core Generator" aracının tek port blok bellek modülü kullanılmıştır. Bunun dışında, ekranda belirlenecek olan görüntü piksellerini okuyacak, VHDL tabanlı bir okuma koduna da ihtiyaç vardır. Gerekli olan bu kod, VGA kontrolör yazılımın haricinde ISE ortamında oluşturulmuş bir alt yazılımdır. Bu kod içerisinde, sayıcı algoritmasıyla üretilen '*addr_normal*' sinyal değişkeni, tek port blok bellek yapısına bağlı konum adreslerini okuması için kullanılmış ve COE uzantılı dosyadan adrese karşılık gelen indeks verileri '*datamem'* sinyalinde tutularak, VGA portunun ilgili pin adreslerine yönlendirilmiştir. Burada COE uzantılı dosya, tek port blok bellek modülünde kullanılmaktadır ve görüntü bilgisine ait olan 3-bit uzunluğunda piksel verilerini içerir. Kaynak görüntüsüyle ilişkili COE dosyası, bu çalışmada MATLAB ortamında oluşturulmuş bir fonksiyon kullanılarak elde edilmiştir.

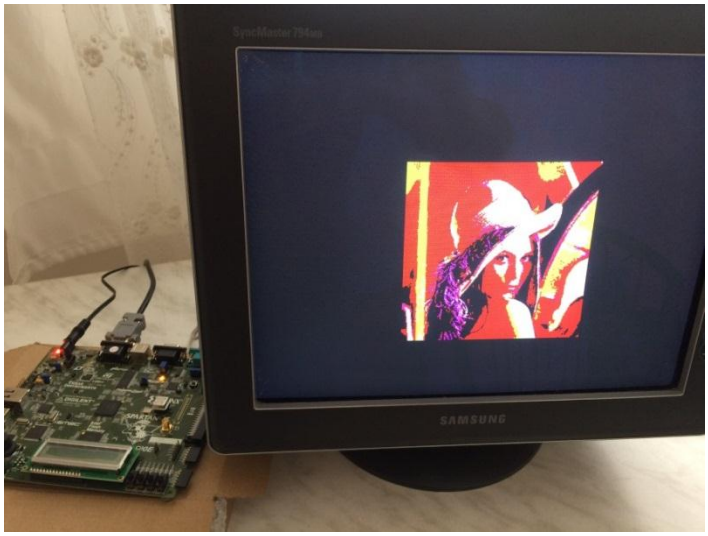
Görüntünün FPGA donanımı üzerinden bir CRT analog monitöre aktarılması noktasında birden fazla test görüntüsü kullanılmıştır. Ayrıca sunulan kripto sistemin uygulama tarafında kaynak görüntünün FPGA ortamında gerçekleştirilen şifreleme sonunda oluşmuş görüntü bilgileri, VGA portu üzerinden başarılı bir şekilde monitör ekranında görüntülenmiştir. Burada veri aktarım işlemi blok bellek yapısı üzerinden sağlanmaktadır. Yani FPGA ortamında şifreleme sonucu oluşan şifrelenmiş veri MATLAB ortamında işlendikten sonra ilk olarak donanımın bellek modülünde tutulmakta daha sonra bu veriler VHDL tabanlı program kümesi ile okunarak görüntünün monitörde oluşması sağlanmaktadır. Uygulama çalışmaların birinde, Şekil 7.24' de görüldüğü gibi 256×224 ölçüsüne sahip 'Mickey Mouse' görüntüsü, FPGA donanımın VGA portu üzerinden başarılı bir şekilde aktarılmıştır. Burada dikkat edilirse bu görüntüyü oluşturan piksellerin tümü sadece siyah, beyaz, sarı, kırmızı ve mavi gibi 3-bit ile temsil edilebilen renk kodlarından oluşmuştur. Dolayısıyla monitörde beliren görüntü, bilgisayarda gözlemlenen orijinal görüntüsüyle aynı kalitededir.

Şekil 7.24 ‘Mickey Mouse’ görüntüsünün FPGA üzerinden aktarılması

Bir başka uygulamada ise 256×256 ölçüsündeki ‘Lena’ görüntüsü benzer şekilde FPGA üzerinden aktarılmıştır. Bununla ilgili uygulama görüntüsü Şekil 7.25’ de verilmiştir. Burada kolayca fark edildiği gibi aktarım sonrası oluşan ‘Lena’ görüntüsü, bilgisayar ortamında gözlemlenen durumundan kalite anlamında düşük seviyededir. Bu durum, beklenilen bir sonuçtur. Çünkü bilgisayar ortamında orijinal ‘Lena’ görüntüsünde her bir piksel 256 farklı renk kodu ile temsil edilirken, FPGA platformunda ise görüntüyü oluşturan pikseller daha önceden belirtildiği gibi ancak 8 farklı renk ile temsil edilebilir. Kısaca bu işlem 8-bit görüntünün, 3-bit bir görüntüye çevrilmesi olarak nitelendirilebilir. Sonuçta, görüntünün aktarılmasından önce bu işlemin yapılması gereklidir. Uygulama çalışmalarında renkli görüntülere yönelik gerçekleştirilen bu işlem, MATLAB ortamında çözümlenmiştir. Burada kullanılan yöntem, renkli görüntünün herbir katmanı için piksel değerinin 127’ den büyük olması durumunda bu değerin 1’ e; küçük olması halinde ise 0’ a kodlanmasına dayalıdır. Bunun sonucunda da görüntü içerisinde belirli bir pozisyondaki pikselin üç katmanı için tek bir RGB bilgisi oluşturulur. Görüntünün tüm piksellerine ait RGB bilgilerinin VGA portuna yönlendirilmesiyle görüntü ekrana yansımaktadır.

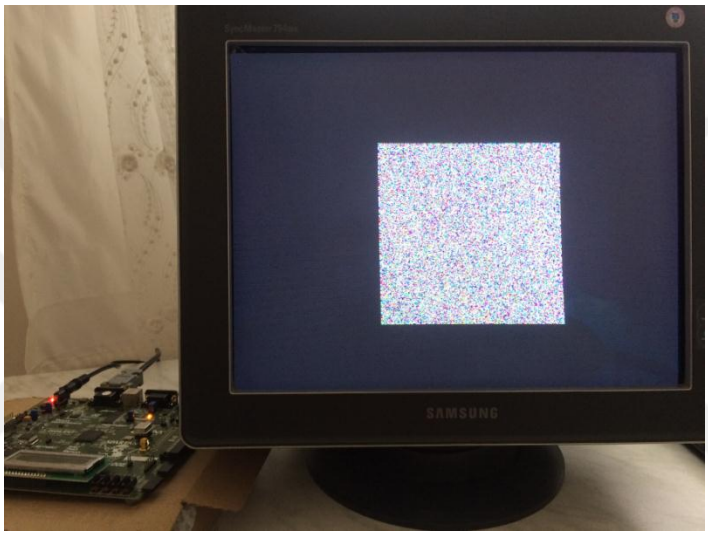
Şekil 7.25 ‘Lena’ görüntüsünün FPGA üzerinden aktarılması

Tasarlanan simetrik görüntü kripto sistemi renkli görüntüler için de uygulanabilir. Gri görüntüler tek katmandan oluşurken; renkli görüntüler ise Kırmızı, Yeşil ve Mavi olmak üzere toplam üç katmandan oluşur. RGB görüntülerin, sunulan kripto sistemde uygulanışında renkli görüntüye ait her bir katman aynı şifreleme algoritmasıyla ayrı ayı şifrelenip, daha sonra şifrelenmiş görüntülerin birleştirilmesiyle şifreleme işlemi tamamlanır. Burada dikkat edilirse, her bir katman için aynı şifreleme parametreleri kullanılsa bile görüntü katmanlarının şifrelenmesinde farklı anahtar kodları kullanılacaktır. Çünkü tasarlanan kripto sistemin kaynak görüntüye karşı duyarlı olması, farklı piksel değerlerine sahip her bir katman için farklı şifreleme kodlarının üretileceği anlamına gelmektedir. Şekil 7.26’ da verilen renkli ‘Lena’ test görüntüsü uygulamalı olarak sunulan şifreleme algoritmasında kaynak görüntü olarak kullanılmıştır.



Şekil 7.26 ‘Lena’ kaynak görüntüsünün monitörde gözlemlenmesi

Renkli görüntünün katmanlarına ayrılması ve şifreleme sonunda birleştirilmesine yönelik işlemler MATLAB ortamında gerçekleştirilirken, her bir katman için şifreleme işlemi ve şifrelenmiş görüntü aktarımı FPGA ortamında sağlanmaktadır. Renkli 'Lena' görüntüsü ait her bir katman, belirli bir şifreleme anahtarı ile kripto sistemde ayrı ayrı şifrelenmiştir. Daha sonra şifrelenmiş her bir katman birleştirilerek, şifrelenmiş 'Lena' görüntüsü oluşturulmuş ve VGA üzerinden hedef monitöre aktarımı başarıyla gerçekleştirilmiştir. Aktarma sonucu oluşan uygulama görüntüsü Şekil 7.27' de verilmiştir.



Şekil 7.27 Şifrelenmiş 'Lena' görüntüsünün monitörde gözlemlenmesi

Sonuç olarak sunulan kripto sistemin FPGA ortamında hem donanımsal benzetimi hem de pratik uygulanabilirliği başarıyla gerçekleştirilmiştir. Uygulamaya yönelik olarak gerçekleştirilen tüm deneysel sonuçlar, kripto sistemin tüm kısımlarının donanım ortamında sorunsuz çalışabilirliğini açıkça göstermektedir. Ayrıca elde edilen teorik ve deneysel sonuçlar, sunulan şifreleme algoritmasına benzer tüm görüntü kripto sistemlerinin pratik alanlarında başarıyla uygulanabileceğini onaylamaktadır.

8. SONUÇLAR VE ÖNERİLER

8.1. Sonuçlar

Bu tez çalışmasında, renkli ve gri görüntüler için kaos tabanlı bir kripto sistem tasarlanmış ve sunulmuştur. Lojistik harita sisteminde mevcut olan sürekli kaos sergilememe özelliği ve kontrol parametresinin sınırlı bir aralığa sahip olması gibi ciddi eksiklikler giderilerek sistem üzerinde önemli iyileştirmeler yapılmıştır. Bu iyileştirmeler neticesinde tasarlanan yeni haritanın hemen hemen tüm kontrol parametrelerinde sürekli kaotik davranış sergilediği, denklem sistemi üzerinde gerçekleştirilen analiz sonuçlarıyla gözlemlenmiştir. Kriptografik anahtar uygunluğuna yönelik olarak gerçekleştirilen tüm istatistiksel testler, yeni sistemden üretilen çıkış serilerinin görüntü şifrelemede kullanılmak üzere yeterli kaliteye sahip olduğunu göstermiştir. Ayrıca iyileştirilen bu harita sisteminin gerçek zamanlı olarak FPGA donanımda başarıyla sentezlenmesi, benzer tüm kaotik sistemlerin sayısal sistemlerde uygulanabilirliğini kanıtlar niteliktedir.

Sunulan şifreleme algoritması kaynak görüntüye bağlı olacak şekilde tasarlanmış, bu da üst seviye bir güvenlik sağlayarak özellikle diferansiyel saldırılara karşı kripto sistemi daha dayanıklı yaptığı, bilgisayar ortamında gerçekleştirilen istatistiksel ve diferansiyel sonuçlarla gözlemlenmiştir. Görüntü şifrelemede kullanılan karıştırma ve dağılma kısımlarının önemi, bu çalışmada elde edilen teorik ve deneysel sonuçlarla birkez daha desteklenmiştir. İyi bir şifreleyicinin, şifrelenecek bilgiye karşı duyarlılık sergilediği düşünüldüğünde, kaynak görüntünün herhangi bir pikselinde bir bit değişimde dahi tamamen farklı şifreleme kodlarının üretildiği ve bunun sonucunda da ortalama %99.59 oranında farklı bir şifreli görüntünün oluştuğu, sunulan şifreleme algoritmasına ait diferansiyel analiz sonuçlarından gözlemlenmiştir. Şifreleme sırasında her bir iterasyonda farklı gizli kodların algoritmada kullanılması, kripto sistemin anahtar alanın arttırılmasında büyük katkı sağladığı görülmüştür. Bununla beraber kripto sistemde kullanılan anahtar kodlarının tespiti için şifreleme algoritmasına yönelik gerçekleştirilen kaba-kuvvet saldırısının başarısız olabilmesi için iterasyonunun en üç kez olması gerektiği görülmüştür.

Kripto sistemle ilgili bu alanda sıkça kullanılan istatistiksel ve diferansiyel analizler yapılmış olup, mümkün mertebede sonuçlar diğer akış şifreleyiciler ile karşılaştırılmıştır. Performans karşılaştırması sonuçlarına göre sunulan kripto sistem, mevcut bazı şifreleme algoritmalarına göre çok daha iyi seviyede olduğu görülmüştür. Ayrıca sunulan görüntü

kripto sistemin iletim sırasında olabilecek veri kaybı ve gürültü saldırısı altındaki performansı incelenmiş, bu şartlar için gerçekleştirilen analiz sonuçlarının tatmin edici ve yeterli seviyede olduğu görülmüştür. Algoritmanın şifreleme ve çözme hızlarının analizinde ise yirmiden fazla test görüntüsü kullanılmış ve elde edilen sonuçlar bilinen bazı şifreleme algoritmaları ile karşılaştırılmıştır. Karşılaştırma sonucu, sunulan şifreleme algoritmasının, hız performansı açısından diğer algoritmalara göre çok daha iyi seviyede olduğu sonucuna varılmıştır.

Uygulama noktasında elde edilen sonuçlar, sunulan şifreleme algoritmasının FPGA ortamındaki başarılı sentezini ve donanımsal benzetiminin hatasız bir şekilde gerçekleştirildiğini göstermektedir. Deneysel gözlemler ise kripto sistemin tüm kısımlarının donanım ortamında sorunsuz çalışabildiğini ve dolayısıyla tasarlanan yapının pratik uygulamalarda kullanılabileceğini açıkça göstermiştir. Bu anlamda, benzer tüm kripto sistemlerin görüntü şifreleme alanında başarıyla uygulanabileceğini söylemek mümkündür.

8.2. Öneriler

Kaos tabanlı görüntü şifreleme algoritmaları için permütasyon aşamasında kaynak bilgiye bağlı bir yapı geliştirilerek istatistiksel ve diferansiyel saldırılara karşı daha dirençli bir kripto sistem geliştirilebilir.

Anahtar alanının arttırılmasına yönelik olarak sistem parametreleri sayıca fazla olan yüksek boyutlu kaotik sistemler sunulan şifreleme algoritmasında kod üreteci olarak kullanılabilir.

Kod üreteci olarak kullanılan İLH sistemi farklı kaotik sistemler ile birleştirilerek daha zengin dinamik davranışlara sahip üreteçler elde edilebilir. Böylelikle fazla sayıda kontrol parametresine bağlı bütünleşik sistemler oluşturulabilir.

Güçlü bir kripto sistem için gerekli olan kaynak bilgiye hassas bağlılık ilkesi için kaynak görüntüye ait piksel pozisyonları tabanlı farklı bir yaklaşım geliştirilebilir veya sunulan algoritmaya ilave edilebilir.

İyileştirilen Lojistik harita sistemiyle birlikte tasarlanan şifreleme algoritması metin veya video gibi farklı multimedya verileri için de kullanılabilir. Bu amaca yönelik olarak, pratiksel anlamda gerçek zamanlı uygulamalar geliştirilebilir.

KAYNAKLAR

[1] **Gao, T.**, **and Chen, Z.**, 2008, Image encryption based on a new total shuffling algorithm, *Chaos, Solitons & Fractals*, **38**, 213-220.

[2] **Kara, M.**, 2013, Siber Saldırılar Siber Savaşlar ve Etkileri, *Yüksek Lisans Programı*, İstanbul Bilgi Üniversitesi, Sosyal Bilimler Enstitüsü, İstanbul.

[3] **Pareek, N. K.**, 2012, Design and analysis of a novel digital image encryption scheme, *International Journal of Network Security & Its Applications*, **4**, 95-108.

[4] **Sakthidasan, K.**, **and Krishna, B. S.**, 2011, A new chaotic algorithm for image encryption and decryption of digital color images, *International Journal of Information and Education Technology*, **1**, 137-141.

[5] **Ahmad, J., and Ahmed, F.**, 2012, Efficiency analysis and security evaluation of image encryption schemes, International Journal of Video & Image Processing and Network Security, **12**, 18-31.

[6] **Farah, T., Hermassi, H., Rhoouma, R., and Belghith, S.**, 2013, Watermarking and encryption scheme to secure multimedia information, *Computer and Information Technology (WCCIT)*, 1-5.

[7] **Younes, M. A. B., and Jantan, A.**, 2008, An image encryption approach using a combination of permutation technique followed by encryption, *International Journal of Computer Science and Network Security*, **8**, 191-197.

[8] **Seyedzadeh, S. M., and Mirzakuchaki, S.**, 2012, A fast color image encryption algorithm based on coupled two-dimensional piecewise chaotic map, *Signal Processing*, **92**, 1202-1215.

[9] **Som, S., and Sen, S.**, 2013, A non-adaptive partial encryption of grayscale images based on chaos, *Procedia Technology*, **10**, 663-671.

[10] **Jolfaei, A., and Mirghadri, A.**, 2011, Image encryption using chaos and block cipher, *Computer and Information Science*, **4**, 172-185.

[11] **Babu, A. M., and Singh, K. J.**, 2013, Performance evaluation of chaotic encryption technique, *American Journal of Applied Sciences*, **10**, 35-41.

[12] **Ye, R.,** 2011, A novel chaos-based image encryption scheme with an efficient permutation-diffusion mechanism, *Optics Communications*, **284**, 5290-5298.

[13] **Fu, C., Chen, J. J., Zou, H., Meng, W. H., Zhan, Y. F., and Yu, Y. W.**, 2012, A chaos-based digital image encryption scheme with an improved diffusion strategy, *Optics Express*, **20**, 2363-2378.

[14] **Hazarika, M.**, 2014, A Review of Chaos Based Image Encryption Techniques, *International Journal of Engineering Research and Technology*, **3**, 2209-2212.

[15] **Gaur, E. A., and Gupta, E. M.**, 2014, Review: Image Encryption Using Chaos Based algorithms, *International Journal of Engineering Research and Applications*, **4**, 904-907.

[16] **Jain, A., Tiwari, N., and Shandilya, M.**, 2014, Image Based Encryption Techniques: A Review, *International Journal of Computer Science and Information Technologies*, **5**, 2886-2889.

[17] **Radwan, A. G., AbdElHaleem, S. H., and Abd-El-Hafiz, S. K.**, 2016, Symmetric encryption algorithms using chaotic and non-chaotic generators: a review, *Journal of Advanced Research*, **7**, 193-208.

[18] **Kant, S.**, 2012, Cryptology and Communication Security, *Defence Science Journal*, **62**, 3-5.

[19] **Alvarez, G., and Li, S.**, 2006, Some basic cryptographic requirements for chaos-based cryptosystems, *International Journal of Bifurcation and Chaos*, **16**, 2129-2151.

[20] **Mrdovic, S., and Perunicic, B.**, 2008, Kerckhoffs' principle for intrusion detection, *Telecommunications Network Strategy and Planning Symposium*,1-8.

[21] **Farajallah, M.**, 2015, Chaos-based crypto and joint crypto-compression systems for images and videos, *PhD Thesis*, Universite De Nantes.

[22] **Vaithyasubramanian, S., Christy, A., and Saravanan, D.**, 2014, An Analysis of Markov Password Against Brute Force Attack for Effective Web Applications, *Applied Mathematical Sciences*, **8**, 5823-5830.

[23] **Zhang, Y., and Xiao, D.**, 2013, Double optical image encryption using discrete Chirikov standard map and chaos-based fractional random transform, *Optics and Lasers in Engineering*, **51**, 472-480.

[24] **Sui, L., Duan, K., Liang, J., Zhang, Z., and Meng, H.**, 2014, Asymmetric multiple-image encryption based on coupled logistic maps in fractional Fourier transform domain, *Optics and Lasers in Engineering*, **62**, 139-152.

[25] http://www.facweb.iitkgp.ernet.in/~sourav/Attacks_on_cryptosystems.pdf Chapter13. 10 Nisan 2017.

[26] **Barukab, O. M., Khan, A. I., Shaik, M. S., Murthy, M. R., and Khan, S. A.**, 2012, Secure Communication using Symmetric and Asymmetric Cryptographic Techniques, *International Journal of Information Engineering and Electronic Business*, **4**, 36-42.

[27] **Sasi, S. B., Dixon, D., Wilson, J., and No, P.**, 2014, A general comparison of symmetric and asymmetric cryptosystems for WSNs and an overview of location based encryption technique for improving security, *IOSR Journal of Engineering*, **4**, 1-4.

[28] **Kuppuswamy, P., and Al-Khalidi, S. Q.**, 2014, Hybrid encryption/decryption technique using new public key and symmetric key algorithm, *International Journal of Information and Computer Security*, **6**, 372-382.

[29] **Menezes, A. J., Van Oorschot, P. C., and Vanstone, S. A.**, 1996, Handbook of Applied Cryptography, *CRC press.*

[30] **Paar, C., and Pelzl, J.**, 2009, Understanding cryptography: a textbook for students and practitioners. *Springer Science & Business Media*.

[31] **Shukla, R., Prakash, H. O., Bhushan, R. P., Venkataraman, S., and Varadan, G.**, 2013, Sampurna Suraksha: unconditionally secure and authenticated one time pad cryptosystem, *International Conference on Machine Intelligence and Research Advancement*, 174-178.

[32] **Tornea, O., Borda, M. E., Pileczki, V., and Malutan, R.**, 2011, DNA Vernam cipher, *International Conference on E-Health and Bioengineering*, 1-4, Romania.

[33] **Jakimoski, G., and Kocarev, L.**, 2001, Chaos and cryptography: block encryption ciphers based on chaotic maps, *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, **48**, 163-169.

[34] **Oğraş, H.**, 2010, Kaos tabanlı sayısal haberleşme sistemlerinin benzetimi için bir grafik kullanıcı arabirim tasarımı, *Yüksek Lisans Tezi*, Fırat Üniversitesi Fen Bilimleri Enstitüsü, Elazığ.

[35] **Hathal, H. M., Abdulhussein, R. A., and Ibrahim, S. K.**, 2014, Lyapunov exponent testing for AWGN Generator system, *Communications and Network*, **6**, 201-208.

[36] **Drake, D. F., and Williams, D. B.**, 1995, Pseudo-chaos for direct-sequence spread-spectrum communication, *International Society for Optics and Photonics*, 104-114.

[37] **Makris, G., and Antoniou, I.**, 2012, Cryptography with chaos, *International Conference* on *Chaotic modelling and Simulation*, 12-15, Greece.

[38] **Shukla, P. K., Khare, A., Rizvi, M. A., Stalin, S., and Kumar, S.**, 2015, Applied cryptography using chaos function for fast digital logic-based systems in ubiquitous computing, *Entropy*, **17**, 1387-1410.

[39] **Sadkhan, S. B., and Mohammed, R. S.**, 2015, Proposed random unified chaotic map as PRBG for voice encryption in wireless communication, *Procedia Computer Science*, **65**, 314-323.

[40] **Prasad, M., and Sudha, K. L.**, 2011, Chaos image encryption using pixel shuffling, *Computer Science & Information Technology*, 169-179.

[41] **Carmen, P. L., and Ricardo, L. R.**, 2012, Notions of chaotic cryptography: sketch of a chaos based cryptosystem, *Applied Cryptography and Network Security*, 267-294.

[42] **Li, S., Alvarez, G., Li, Z., and Halang, W. A.**, 2007, Analog chaos-based secure communications and cryptanalysis: A brief survey, *3rd International IEEE Scientific Conference on Physics and Control*, Germany.

[43] **Kamil, I. A., and Fakolujo, O. A.**, 2012, Lorenz-based chaotic secure communication schemes, *Ubiquitous Computing and Communication Journal*, **7**, 1248-1254.

[44] **Fridrich, J.**, 1998, Symmetric ciphers based on two-dimensional chaotic maps, *International Journal of Bifurcation and Chaos*, **8**, 1259-1284.

[45] **Li, S., and Zheng, X.**, 2002, Cryptanalysis of a chaotic image encryption method, *IEEE International Symposium on Circuits and Systems*, **2**,708-711.

[46] **Yen, J. C., and Guo, J. I.**, **2000**, A new chaotic key-based design for image encryption and decryption, *IEEE International Symposium on Circuits and Systems*, **4**, 49-52.

[47] **Shujun, L., Xuanqin, M., and Yuanlong, C.**, 2001, Pseudo-random bit generator based on couple chaotic systems and its applications in stream-cipher cryptography, *Progress in Cryptology*, **2247**, 316-329.

[48] **Gao, H., Zhang, Y., Liang, S., and Li, D.**, 2006, A new chaotic algorithm for image encryption, *Chaos, Solitons & Fractals*, **29**, 393-399.

[49] **Pan, J., Ding, Q., and Qi, N.**, 2012, The Research of Chaos-based SMS Encryption in Mobile Phone, *Second International Conference on Instrumentation & Measurement, Computer, Communication and Control*, 501-504.

[50] **Chen, C. K., and Lin, C. L.,** 2010, Text encryption using ECG signals with chaotic Logistic map, *5th IEEE Conference on Industrial Electronics and Applications*, 1741-1746.

[51] **Vaferi, E., and Sabbaghi-Nadooshan, R.**, 2015, A new encryption algorithm for color images based on total chaotic shuffling scheme, *Optik-International Journal for Light and Electron Optics*, **126**, 2474-2480.

[52] **Li, J., Zhong, T., Jiang, M., Dai, B., He, R., and Li, R.**, 2016, Digital camera with image encryption, *Optik-International Journal for Light and Electron Optics*, **127**, 1391-1394.

[53] **Xu, L., Li, Z., Li, J., and Hua, W.**, 2016, A novel bit-level image encryption algorithm based on chaotic maps, *Optics and Lasers in Engineering*, **78**, 17-25.

[54] **Li, X., Li, C., and Lee, I. K.**, 2016, Chaotic image encryption using pseudo-random masks and pixel mapping, *Signal Processing*, **125**, 48-63.

[55] **Gu, G., Ling, J., Xie, G., and Li, Z.**, 2016, A chaotic-cipher-based packet body encryption algorithm for JPEG2000 images, *Signal Processing: Image Communication*, **40**, 52-64.

[56] **Norouzi, B., Seyedzadeh, S. M., Mirzakuchaki, S., and Mosavi, M. R**., 2015, A novel image encryption based on row-column, masking and main diffusion processes with hyper chaos, *Multimedia Tools and Applications*, **74**, 781-811.

[57] **Chen, J. X., Zhu, Z. L., Fu, C., and Yu, H.**, 2013, An improved permutation-diffusion type image cipher with a chaotic orbit perturbing mechanism, *Optics Express*, **21**, 27873-27890.

[58] **Yoon, J. W., and Kim, H.,** 2010, An image encryption scheme with a pseudorandom permutation based on chaotic maps, *Communications in Nonlinear Science and Numerical Simulation*, **15**, 3998-4006.

[59] **Ismail, I. A., Amin, M., and Diab, H.**, 2010, A digital image encryption algorithm based a composition of two chaotic logistic maps, *International Journal of Network Security*, **11**, 1-10.

[60] **Musa, M. A., Schaefer, E. F., and Wedig, S.**, 2003, A simplified AES algorithm and its linear and differential cryptanalyses, *Cryptologia*, **27**, 148-177.

[61] **Nayak, C. K., Acharya, A. K., and Das, S.,** 2011, Image encryption using an enhanced block based transformation algorithm, *International Journal of Research and Reviews in Computer Science*, **2**, 275-279.

[62] **Ma, X., Fu, C., Lei, W. M., and Li, S.**, 2011, A novel chaos-based image encryption scheme with an improved permutation process, *International Journal of Advancements in Computing Technology*, **3**, 223-233.

[63] **Peng, J., Zhang, D., and Liao, X.**, 2013, A novel algorithm for block encryption of digital image based on chaos, *International Journal of Cognitive Informatics and Natural Intelligence*, **5**, 59-74.

[64] **Sathishkumar, G. A., Bagan, K. B., and Sriraam, D. N.**, 2011, Image encryption based on diffusion and multiple chaotic maps, *International Journal of Network Security & Its Applications*, **3**, 181-194.

[65] **Chen, G., Mao, Y., and Chui, C. K.,** 2004, A symmetric image encryption scheme based on 3D chaotic cat maps, *Chaos, Solitons & Fractals*, **21**, 749-761.

[66] **Solak, E., Çokal, C., Yildiz, O. T., and Bıyıkoğlu, T.,** 2010, Cryptanalysis of Fridrich's chaotic image encryption, *International Journal of Bifurcation and Chaos*, **20**, 1405-1413.

[67] **Behnia, S., Akhshani, A., Mahmodi, H., and Akhavan, A.**, 2008, Chaotic cryptographic scheme based on composition maps, *International Journal of Bifurcation and Chaos*, **18**, 251-261.

[68] **Li, C., Arroyo, D., and Lo, K. T.**, 2010, Breaking a chaotic cryptographic scheme based on composition maps, *International Journal of Bifurcation and Chaos*, **20**, 2561-2568.

[69] **Kanso, A., and Ghebleh, M.**, 2012, A novel image encryption algorithm based on a 3D chaotic map, *Communications in Nonlinear Science and Numerical Simulation*, **17**, 2943-2959.

[70] **Lian, S., Sun, J., and Wang, Z.**, 2005, A block cipher based on a suitable use of the chaotic standard map, *Chaos, Solitons & Fractals*, **26**, 117-129.

[71] **Wong, K. W., Kwok, B. S. H., and Law, W. S.**, 2008, A fast image encryption scheme based on chaotic standard map, *Physics Letters A*, **372**, 2645-2652.

[72] **Wang, Y., Wong, K. W., Liao, X., Xiang, T., and Chen, G.**, 2009, A chaos-based image encryption algorithm with variable control parameters, *Chaos, Solitons & Fractals*, **41**, 1773-1783.

[73] **Zhu, C.**, 2012, A novel image encryption scheme based on improved hyperchaotic sequences, *Optics Communications*, **285**, 29-37.

[74] **Pisarchik, A. N., and Zanin, M.**, 2008, Image encryption with chaotically coupled chaotic maps, *Physica D: Nonlinear Phenomena*, **237**, 2638-2648.

[75] **Fu, C., Lin, B. B., Miao, Y. S., Liu, X., and Chen, J. J.**, 2011, A novel chaos-based bit-level permutation scheme for digital image encryption, *Optics Communications*, **284**, 5415-5423.

[76] **Zhu, Z. L., Zhang, W., Wong, K. W., and Yu, H.**, 2011, A chaos-based symmetric image encryption scheme using a bit-level permutation, *Information Sciences*, **181**, 1171-1186.

[77] **Wu, Y., Agaian, S., and Noonan, J. P.**, 2012, A new family of generalized 3D cat maps, *arXiv:1205.3208*.

[78] **Phatak, S. C., and Rao, S. S.,** 1995, Logistic map: A possible random-number generator, *Physical review E*, **51**, 3670-3678.

[79] **Tricarico, M., and Visentin, F.**, 2014, Logistic map: from order to chaos, *Applied Mathematical Sciences*, **8**, 6819-6826.

[80] **Patidar, V., Sud, K. K., and Pareek, N. K.**, 2009, A pseudo random bit generator based on chaotic logistic map and its statistical testing, *Informatica*, **33**, 441-452.

[81] **Murillo-Escobar, M. A., Cruz-Hernández, C., Abundiz-Pérez, F., López-Gutiérrez, R. M., and Del Campo, O. A.**, 2015, A RGB image encryption algorithm based on total plain image characteristics and chaos, *Signal Processing*, **109**, 119-131.

[82] **Zhou, Y., Bao, L., and Chen, C. P.**, 2014, A new 1D chaotic system for image encryption, *Signal Processing*, **97**, 172-182.

[83] **Lang, J., Tao, R., and Wang, Y.**, 2010, Image encryption based on the multiple-parameter discrete fractional Fourier transform and chaos function, *Optics Communications*, **283**, 2092-2096.

[84] **Wang, X., Zhao, J., and Liu, H.**, 2012, A new image encryption algorithm based on chaos, *Optics Communications*, **285**, 562-566.

[85] **Wong, K. W., Kwok, B. S. H., & Yuen, C. H.,** 2009, An efficient diffusion approach for chaos-based image encryption, *Chaos, Solitons & Fractals*, **41**, 2652-2663.

[86] **Kumar, R. R., and Kumar, M. B.**, 2014, A new chaotic image encryption using parametric switching based permutation and diffusion, *Journal on Image and Video Processing*, **4**, 795-804.

[87] **Jani, J., and Malkaj, P.**, 2014, Numerical Calculation of Lyapunov Exponents in Various Nonlinear Chaotic Systems, *International Journal of Scientific & Technology Research*, **3**, 87-90.

[88] **Eisencraft, M., and Kato, D. M.**, 2009, Spectral properties of chaotic signals with applications in communications, *Nonlinear Analysis: Theory, Methods & Applications*, **71**, 2592-2599.

[89] **François, M., Defour, D., and Negre, C.**, 2014, A fast Chaos-based pseudo-random bit generator using binary64 floating-point arithmetic, *Informatica*, **38**, 115-124.

[90] **Li, C. Y., Chen, J. S., and Chang, T. Y.**, 2006, A chaos-based pseudo random number generator using timing-based reseeding method, *IEEE International Symposium on Circuits and Systems*, 3277-3280.

[91] **Min, L., Chen, T., and Zang, H.**, 2013, Analysis of FIPS140-2 test and chaos-based pseudorandom number generator, *Chaotic Modeling and Simulation*, **2**, 273-280.

[92] **Gupta, K., and Silakari, S.**, 2011, New approach for fast color image encryption using chaotic map, *Journal of Information Security*, **2**, 139-150.

[93] **Zaman, J. K. M. S., and Ghosh, R.**, 2012, Review on fifteen Statistical Tests proposed by NIST, *Journal of Theoretical Physics and Cryptography*, **1**, 18-31.

[94] **Xiangdong, L. I. U., Junxing, Z., Jinhai, Z., and Xiqin, H.**, 2008, Image scrambling algorithm based on chaos theory and sorting transformation, *International Journal of Computer Science and Network Security*, **8**, 64-68.

[95] **Fu, C., Huang, J. B., Wang, N. N., Hou, Q. B., and Lei, W. M.**, 2014, A symmetric chaos-based image cipher with an improved bit-level permutation strategy, *Entropy*, **16**, 770-788.

[96] **Boriga, R., Dascalescu, A. C., and Mihailescu, M. I.**, 2014, A novel chaos-based image encryption scheme, *Annals of the University of Craiova-Mathematics and Computer Science Series*, **41**, 47-58.

[97] **Zhang, L. Y., Hu, X., Liu, Y., Wong, K. W., and Gan, J**., 2014, A chaotic image encryption scheme owning temp-value feedback, *Communications in Nonlinear Science and Numerical Simulation*, **19**, 3653-3659.

[98] **Ahmad, M., and Al-Sharari, H. D.,** 2014, An Inter-Component pixels permutation based color image encryption using hyper-chaos, *arXiv:1403.4780*.

[99] **Guo, W., Zhao, J., and Ye, R.**, 2014, A chaos-based pseudorandom permutation and bilateral diffusion scheme for image encryption, *International Journal of Image, Graphics and Signal Processing*, **6**, 50-61.

[100] **Guanghui, C., Kai, H., Yizhi, Z., Jun, Z., and Xing, Z.**, 2014, Chaotic image encryption based on running-key related to plain-text, *The Scientific World Journal*, Article ID 490179.

[101] **Cao, G. H., Zhou, J., and Zhang, Y. Z.**, 2014, Quantum chaotic image encryption with one time running key, *International Journal of Security and Its Applications*, **8**, 77-88.

[102] **Zhou, S., Wei, Z., Wang, B., Zheng, X., Zhou, C., and Zhang, Q.**, 2016, Encryption method based on a new secret key algorithm for color images, *International Journal of Electronics and Communications*, **70**, 1-7.

[103] **Noura, H., Sleem, L., and Couturier, R.**, 2017, A Revision of a New Chaos-Based Image Encryption System: Weaknesses and Limitations, *arXiv:1701.08371*.

[104] **Bahrami, S., and Naderi, M.**, 2012, Image encryption using a lightweight stream encryption algorithm, *Advances in Multimedia*, Article ID 767364.

[105] **Bokhari, M. U., Alam, S., and Masoodi, F. S.**, 2012, Cryptanalysis techniques for stream cipher: a survey, *International Journal of Computer Applications*, **60**, 29-33.

[106] **Kendhe, A. K., and Agrawal, H.,** 2013, A Survey Report on Various Cryptanalysis Techniques, *International Journal of Soft Computing and Engineering*, **3**, 287-293.

[107] **Hauck, S.,** 1998, The roles of FPGAs in reprogrammable systems, *Proceedings of the IEEE*, **86**, 615-638.

[108] **Saidani, T., Dia, D., Elhamzi, W., Atri, M., and Tourki, R.**, 2009, Hardware co-simulation for video processing using Xilinx System Generator, *Proceedings of the World Congress on Engineering*, **1**, 3-7.

[109] **Radi, H. R., Caleb, W. W. K., Zainudin, M. N. S., Ismail, M. M.,** 2012, The design and implementation of VGA controller on FPGA, *International Journal of Electrical & Computer Sciences*, **12**, 56-60.

ÖZGEÇMİŞ

Yazar, 1983 yılında Diyarbakır Merkez ilçesinde doğdu. İlköğrenimini, Cemil Özgür ilköğretim okulunda tamamladıktan sonra ortaöğrenim ve liseyi, Nevzat Ayaz Anadolu lisesinde okuyarak 2001 yılında bu okuldan mezun oldu.

2002 yılında girdiği Gaziantep Üniversitesi, Mühendislik Fakültesi, Elektrik-Elektronik Mühendisliğini bölümünden 2008 yılında mezun oldu. Yazar, 2007 yılında eğitimini aldığı TURKCELL-GSM akademi belgesi ile aynı yılda Turkcell-Taksim plazada bilgisayar ağları üzerine 2 aylık bir staj deneyimi kazanmıştır. Üniversiteden mezun olduktan sonra HUAWEI-TURKCELL işbirliği projesi kapsamında, Baz istasyonlarının donanım ve yazılım güncellemesine yönelik 3 ay iş tecrübesi kazandıktan sonra 2009 yılında Batman Üniversitesi, Teknik Eğitim Fakültesi, Elektrik Eğitimi bölümünde Araştırma Görevlisi olarak göreve başladı. 2008 yılında başladığı, Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik-Elektronik Mühendisliği anabilim dalındaki yüksek lisansını 2010 yılında tamamladı. Aynı yıl Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik-Elektronik Mühendisliği anabilim dalı, Telekomünikasyon bilim dalında doktora eğitimine başladı.

Batman Üniversitesinde halen Araştırma Görevlisi kadrosunda çalışan yazar, evli ve bir çocuk babasıdır.

Elektronik Posta: hidayet.ogras@batman.edu.tr