T.C.
İSTANBUL ÜNİVERSİTESİ-CERRAHPAŞA
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

DOKTORA TEZİ

KAOS TABANLI HAFİF SIKLET BİR ŞİFRELEME ALGORİTMASININ TASARLANMASI ve UYGULANMASI

Yasin KAYA

DANIŞMAN
Dr. Öğr. Üyesi Gülsüm Zeynep GÜRKAŞ AYDIN

II. DANIŞMAN
Prof. Dr. Akif AKGÜL

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği, Doktora Programı

Nisan, 2025

TEZ KABUL VE ONAYI

Yasin KAYA tarafından, **Dr. Öğr. Üyesi Gülsüm Zeynep GÜRKAŞ AYDIN** danışmanlığında hazırlanan **" KAOS TABANLI HAFİF SIKLET BİR ŞİFRELEME ALGORİTMASININ TASARLANMASI ve UYGULANMASI "** başlıklı bu çalışma, jürimiz tarafından …. tarihinde yapılan sınav sonucunda **……..** ile başarılı bulunarak **Doktora Tezi** olarak kabul edilmiştir.

Tez Jürisi

		İmza	**Sonuç**
DANIŞMAN	Dr. Öğr. Üyesi Gülsüm Zeynep GÜRKAŞ AYDIN İstanbul Üniversitesi-Cerrahpaşa Bilgisayar Mühendisliği Anabilim Dalı		☐ Kabul ☐ Ret
ÜYE	Prof. Dr. Zeynep ORMAN İstanbul Üniversitesi-Cerrahpaşa Bilgisayar Mühendisliği Anabilim Dalı		☐ Kabul ☐ Ret
ÜYE	Dr. Öğr. Üyesi Ali BOYACI İstanbul Ticaret Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı		☐ Kabul ☐ Ret
ÜYE	Prof. Dr. Ahmet SERTBAŞ İstanbul Üniversitesi-Cerrahpaşa Bilgisayar Mühendisliği Anabilim Dalı		☐ Kabul ☐ Ret
ÜYE	Doç. Dr. Can EYÜPOĞLU Milli Savunma Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı		☐ Kabul ☐ Ret

Aileme ithaf ediyorum…

BÜTÇE DESTEKLERİ

KAOS TABANLI HAFİF SIKLET BİR ŞİFRELEME ALGORİTMASININ TASARLANMASI ve UYGULANMASI

Bu tez çalışması için herhangi bir kurumdan bütçe desteği alınmamıştır.

TEŞEKKÜR

Doktora tez çalışmam sürecinde tecrübe ve bilgileriyle her zaman bana büyük destek olan danışman hocalarım Dr. Öğr. Üyesi Gülsüm Zeynep GÜRKAŞ AYDIN'a ve Prof. Dr. Akif AKGÜL'e, tez izle komitesinde bulunup değerli fikirleri ile katkıda bulunan Prof. Dr. Zeynep ORMAN'a, Dr. Öğr. Üyesi Ali BOYACI'ya çok teşekkür ediyorum.

Nisan 2025 Yasin KAYA

İÇİNDEKİLER

Sayfa No

ŞEKİL LİSTESİ

Sayfa No

TABLO LİSTESİ

Sayfa No

SİMGE VE KISALTMA LİSTESİ

Simgeler **Açıklama**

P : Plaintext – Açık metin
C : Chipertext – Şifreli metin
K : Key – Anahtar
H : Hash Code – Özet kod
H2 : Özet kodun ilk iki byte'ı
HV3 : Özet kodun ilk iki byte'ı ile validasyon byte'ını içeren 3 byte'lık veri
R2 : İki byte'lık rastgele değer
HV3 : İki byte'lık rastgele değer ile validasyon byte'ını içeren 3 byte'lık veri
n : Satır Sayısı
m : Sütun sayısı
i7 : Intel Core i7 CPU

Kısaltmalar **Açıklama**

AES : Advanced Encryption Standard – Gelişmiş Şifreleme Standardı
DES : Data Encryption Standard – Veri Şifreleme Standardı
3DES : Triple Data Encryption Standard – Üçlü Veri Şifreleme Standardı
RSA : Rivest-Shamir-Adleman
DSA : Digital Signature Algorithm – Dijital İmza Algoritması
ECC : Elliptic-Curve Cryptography – Eliptik Eğri Kriptografi
CPU : Central Process Unit – Merkezi İşlem Birim
RAM : Random Access Memory – Rastgele Erişimli Bellek
IoT : Internet of Things – Nesnelerin İnterneti
TDES : Triple Data Encryption Standard – Üçlü Veri Şifreleme Standardı
RC4 : Rivest Cipher 4 – Rivest Şifresi 4
RC5 : Rivest Cipher 5 – Rivest Şifresi 5
DSA : Digital Signature Algorithm - Dijital İmza Algoritması
EC : Elliptic Curve - Eliptik Eğri
TLS : Transport Layer Security - Taşıyıcı Katman Güvenliği

LWC	: Lightweight Cryptography – Hafif Sıklet Kriptografi
FN	: Feistel Networks – Feistel Ağlar
SPN	: Substitution-Permutation Networks – Yerine koyma-Yerdeğiştirme Ağları
S-BOX	: Substitution-box – Yerine koyma kutusu
PSO	: Particle Swarm Optimization – Parçacık Sürü Optimizasyonu
DNA	: Deoxyribonucleic Acid – Deoksiriboz Nükleik Asit
SHA	: Secure Hash Algorithm – Güvenli Özet Algoritması
RCP	: Row Circular Permutation – Satır Tabanlı Dairesel Yer Değiştirme
CCP	: Colum Circular Permutation – Sütun Tabanlı Dairesel Yer Değiştirme
CRC	: Cyclic Redundancy Check – Döngüsel Artıklık Kontrolü
XoR	: Exclusive-OR – Özel OR
GANs	: Adversarial Networks – Karşıt Ağlar
PSNR	: Peak Signal-to-Noise Ratio – En Yüksek Sinyal-Gürültü Oranı
PNG	: Portable Network Graphic – Taşınabilir Ağ Grafiği
NPCR	: Number of Pixels Change Rate – Değişikliğe Uğramış Piksel Oranı
UACI	: Unified Average Changing Intensity – Birleşik Ortalama Değişim Yoğunluğu
MSE	: Mean Square Error – Ortalama Karesel Hata
OS	: Operation System – İşletim Sistemi
GiB	: Gigabyte
TEA	: Tiny Encryption Algorithm – Küçük Şifreleme Algoritması
xTEA	: eXtended TEA – Genişletilmiş TEA
HIGTH	: HIGh security and light weigHT – Yüksek Güvenlik ve Hafif Sıklet
LEA	: Lightweight Encryption Algorithm – Hafif Sıklet Şifreleme Algoritması
RDH	: Reversible Data Hiding – Geri Dönüşümlü Veri Saklama
RFID	: Radio-Frequency IDentification – Radyo Frekansı Tanımlama
FPGA	: Field-Programmable Gate Array – Programlanabilir Kapı Dizileri
PRNG	: Pseudo-Random Number Generator – Sahte Rastgele Sayı Üreteci
TRNG	: True-Random Number Generator – Gerçek Rastgele Sayı Üreteci

ÖZET

DOKTORA TEZİ

KAOS TABANLI HAFİF SIKLET BİR ŞİFRELEME ALGORİTMASININ TASARLANMASI ve UYGULANMASI

Yasin KAYA

İstanbul Üniversitesi-Cerrahpaşa

Lisansüstü Eğitim Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği, Doktora Programı

Danışman : Dr. Öğr. Üyesi Gülsüm Zeynep GÜRKAŞ AYDIN
II. Danışman : Prof. Dr. Akif AKGÜL

Hızla dijitalleşen dünyamızda veri güvenliği her geçen gün daha büyük önem kazanmaktadır. AES, DES, RSA gibi birçok şifreleme algoritması güvenlik anlamında tatmin edici sonuçlar sunmaktadır. Fakat bu algoritmalar kompleks yapıları ve yüksek kaynak (CPU, RAM, enerji vs.) tüketimi nedenleriyle IoT cihazlar, kablosuz sensor ağlar, RFID sistemler gibi kaynak kısıtlı platformlar için uygun değildir. Bu tez çalışması kapsamında kaynak kısıtlı platformların veri güvenliğini sağlayacak hafif sıklet bir şifreleme algoritması geliştirilmiştir.

Geliştirilen bu şifreleme algoritmasında rastgele sayı üreteci olarak iki farklı lojistik haritadan yararlanılmıştır. İlk haritanın başlangıç parametreleri doğrudan gizli anahtardan türetilmiş ve elde edilen çıktı karıştırma (confusion) aşamasında kullanılmıştır. İkinci haritanın başlangıç parametreleri gizli anahtar ve orijinal dosyaya ait özet kod (hash code) kullanılarak belirlenmiş ve elde edilen çıktı yayılma (diffusion) aşamasında kullanılmıştır. Bu sayede yaklaşık 196 bit anahtar uzunluğu ile kaba kuvvet saldırılarına (brute force attacks) karşı dirençli ve 7.9970 üzeri bilgi entropi değeri ile yüksek tahmin edilemezlik (unpredictability) seviyesine sahip bir algoritma sunulmuştur. Yayılma aşamasında ikinci kaotik haritanın kullanılması 99.6% üzerinde NPCR ve 33% civarında UACI değerlerine ulaşılmasını

sağlamıştır ki bu durum diferansiyel saldırılara karşı önerilen algoritmanın direncini göstermektedir. Satır ve sütun bazlı döngüsel rotasyon (cyclic rotation) işlemleri algoritmanın kırpma saldırılarına karşı direncini arttırmıştır.

Güvenlik ve performans testleri, geliştirilen algoritmanın benzerlerine göre ortalamaların üzerinde güvenlik sunduğunu, hız testleri muadillerinden çok daha hızlı olduğunu, karmaşıklık analizi ise oldukça sade ve basit bir yapıda olduğunu göstermiştir. Bütün bu sonuçlar tez kapsamında sunulan algoritmanın kaynak kısıtlı sistemler için güçlü bir alternatif olduğunu göstermektedir. ⌋

Nisan 2025 , ⌊107⌋ sayfa.
Anahtar kelimeler: ⌊ Doğrusal Olmayan Dinamikler, Kaos, Kriptografi, Hafif Sıklet, Resim Şifreleme ⌋

ABSTRACT

Ph.D. THESIS

DESIGN AND IMPLEMENTATION OF A CHAOS-BASED LIGHTWEIGHT ENCRYPTION ALGORITHM

Yasin KAYA

İstanbul University-Cerrahpaşa

Institute of Graduate Studies

Department of Computer Engineering

Computer Engineering Programme

Supervisor : Assist. Prof. Dr. Gülsüm Zeynep GÜRKAŞ AYDIN
Co-Supervisor: Prof. Dr. Akif AKGÜL

In our rapidly digitalizing world, data security is becoming more and more important every day. Many encryption algorithms such as AES, DES, RSA provide satisfactory results in terms of security. However, due to their complex structures and high resource (CPU, RAM, energy, etc.) consumption, these algorithms are not suitable for resource constrained platforms such as IoT devices, wireless sensor networks, RFID systems. Within the scope of this thesis, a lightweight encryption algorithm has been developed to ensure data security for resource-constrained platforms.

In this developed encryption algorithm, two different logistic maps were employed as random number generators. The initial parameters of the first map were directly derived from the secret key and the obtained output was used in the confusion phase. The initial parameters of the second map were determined using the secret key and the hash code of the original file and the obtained output was used in the diffusion phase. In this way, an algorithm that is resistant to brute force attacks with a key length of approximately 196 bits and has a high unpredictability level with an information entropy value of over 7.9970 was presented. The use of the second chaotic map in the diffusion phase provided NPCR values over 99.6% and UACI

values around 33%, which shows the resistance of the proposed algorithm against differential attacks. Row and column based cyclic rotation operations increased the resistance of the algorithm against cropping attacks.

Security and performance tests have shown that the developed algorithm offers above average security compared to its peers, speed tests have illustrated that it is much faster than its counterparts, and complexity analysis has presented that it has a very plain and simple structure. All these results show that the algorithm presented within the scope of the thesis is a strong alternative for resource-constrained systems.

April 2025, [107] pages.
Keywords: [Nonlinear Dynamics, Chaos, Cryptography, Lightweight, Image Encryption]

1. GİRİŞ

Dijital dünyada yaşanan büyük gelişmeler veri güvenliğinin önemini daha fazla akıllara getirmektedir. Her ne kadar bilgi güvenliği teknolojinin gelişmesine paralel olarak son zamanlarda artan hızda hayatımıza girmiş olsa da eski Roma'dan Antik Yunan'a hatta eski Mısır'a kadar tarihin her safhasında karşımıza çıkmaktadır. Antik Yunanda kullanılan Scytale of Sparta, Roma da kullanılan Caesar şifresi bunların en bilinen örnekleridir [1].

Kısaca şifre bilimi olarak isimlendirilen kriptoloji şifreleme tekniklerini içeren kriptografi ve yetkisiz şifre kırma tekniklerini içeren kriptanaliz olmak üzere iki alt disipline ayrılmıştır. Kriptografi, bir mesajın yetkisiz kişiler ya da sistemler tarafından anlaşılamayacak bir forma dönüştürülmesi ve bu şifrelenmiş olan veriden yetkili kişilerin ya da sistemlerin tekrar orijinal mesajı elde edilmesi tekniklerini inceler. Kriptanaliz ise şifrelenmiş bir veriden, yetkisiz bir şekilde, anahtar kullanmadan orijinal veriye ulaşma yani şifre kırma tekniklerini inceler.

Kriptografi, kullanılan teknikler bakımında simetrik şifreleme, asimetrik şifreleme ve protokoller olmak üzere 3 alt dala ayrılmıştır. Simetrik şifrelemede veri şifrelemek için kullanılan anahtar ile şifrelenmiş veriden orijinal veriyi elde etmek için kullanılan anahtarlar aynıdır. Asimetrik şifrelemede ise bu anahtarlar farklıdır. Günümüz popüler şifreleme algoritmalarına baktığımızda karşımıza tatmin edici sonuçlar sunan ve güvenilirliği uygun bulunmuş birçok şifreleme algoritmasından söz etmek mümkündür. Bazı popüler şifreleme algoritmaları şunlardır;

- AES (Advanced Encryption Standard – Gelişmiş Şifreleme Standardı)
- DES (Data Encryption Standard – Veri Şifreleme Standardı)
- 3DES (Triple Data Encryption Standard – Üçlü Veri Şifreleme Standardı)
- Blowfish
- RSA (Rivest-Shamir-Adleman)

Günden güne dijital veri hayatımızın her alanına girmektedir. Giyilebilir cihazlar (wearable devices), akıllı evler (smart home), kablosuz sensor ağlar (wireless sensor network), nesnelerin interneti (internet of things - IoT), RFID tag (Radio-frequency identification) vs. bunlardan bazılarıdır. Bu cihazların ortak özelliği yapısal kaynak kısıtları olmasıdır. Bu kaynaklar, işlemci (CPU), hafıza (RAM), veri depolama alanı (storage), veri yolu genişliği (bandwidth), enerji olarak sıralanabilir.

Yukarıda bahsetmiş olduğumuz günümüz kabul görmüş şifreleme algoritmaları veri güvenliğini ön plana almakta olup karmaşık algoritma yapılarına sahiptirler. Bu karmaşık yapı yüksek kaynak kullanımına ihtiyaç duymakta olup kaynak kısıtlı sistemler için bu algoritmaları elverişsiz hale getirmektedir. Bu soruna yönelik son zamanlarda birçok çalışma yapıldığından bahsedebiliriz [2-22]. Genel olarak hafif sıklet şifreleme algoritmaları (lightweight encryption algorithms) olarak isimlendirilen bu tarz şifreleme algoritmaları, veri güvenliğini maksimize ederken kaynak ihtiyacını minimize etmeyi amaçlamaktadırlar. Bu sayede kaynak kısıtlı sistemlerde veri güveliği sağlanabilmekte ve kaynaklar israf edilmemektedir.

Son zamanlarda yayınlanan hafif sıklet şifreleme algoritmalarına baktığımızda kaotik sitemlerin şifreleme algoritmalarında yoğun bir şekilde kullanıldığını gözlemlemekteyiz. Kaos tabanlı sistemler doğuştan sahip oldukları çeşitli özellikler ile şifreleme algoritmaları için çok elverişli bir altyapı sunmaktadırlar. Kaotik sistemlerin bu özelliklerini şu şekilde sıralamak mümkündür [23];

- İlk durum parametrelerine hassas bağlılık (sensitive to initial conditions)
- Tekrarlanmayan çıktı (non-repeating)
- Çıktının belirli bir aralıkta olması (bounded)
- Formüle edilebilir kararlı yapı (deterministic)
- Doğrusal olmama (non-linearity)

Büyük bir verinin kısa kimliğini oluşturmada kullanılan algoritmalara özet fonksiyonlar (hash functions) denir. Bu özet fonksiyonlar girdi olarak aldıkları büyük veriden sabit uzunlukta (örneğin 128 bit, 256 bit vs.) bir özet kod (hash code/hash value) üretirler ve bu özet kod girdiye sıkı bağlılık içerir. Girdi üzerinde yapılacak bir bitlik değişiklik bile çok farklı bir özet değerin

üretilmesine sebep olur. Bu özelliklerinden dolayı özet fonksiyonlar çeşitli şifreleme algoritmalarında kullanılmaktadırlar [24, 25]. Bu tez çalışması kapsamında özet fonksiyonlardan yararlanılmıştır.

Tez kapsamında önerdiğimiz ve geliştirdiğimiz kaos tabanlı hafif sıklet şifreleme algoritması ile kaynak kısıtlı sistemlerin güvenlik gereksinimlerini karşılamak amaçlanmıştır. Mevcut çalışmalardan tamamen farklı ve özgün olarak tasarlayıp geliştirdiğimiz algoritmamızda kaotik haritaların yanı sıra özet fonksiyonlar ve çeşitli matematiksel modeller kullanılmıştır. Gerek bulgular gerekse mevcut diğer çalışmalar ile olan karşılaştırmalar geliştirmiş olduğumuz bu özgün algoritmanın oldukça hızlı ve güvenli olduğunu göstermektedir. Test ve analiz sonuçları geliştirilen şifreleme algoritmasının kaynak kısıtlı sistemler için başarılı bir güvenlik algoritması alternatifi olduğunu göstermiştir.

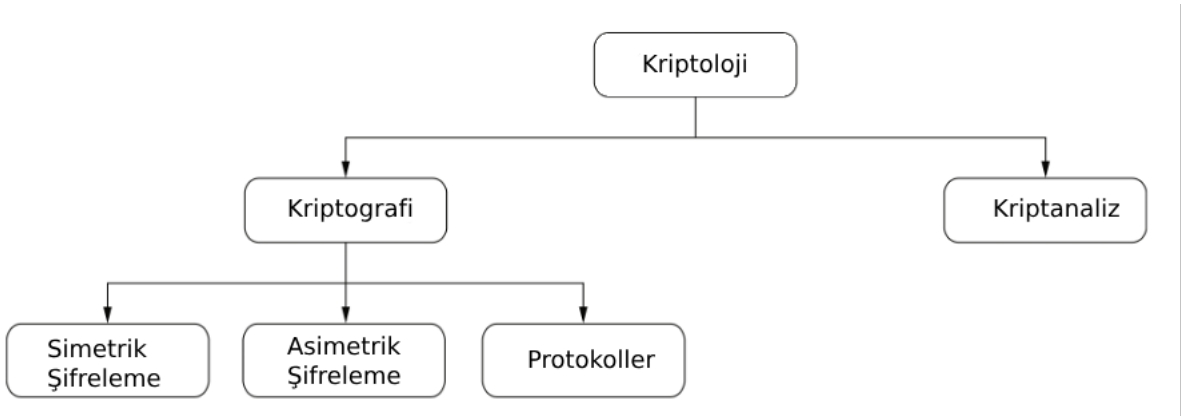
Bu tez çalışmasının geri kalan kısmı şu şekilde organize edilmiştir: Bölüm 2'de şifreleme algoritmaları, hafif sıklet şifreleme algoritmaları, kaotik haritalar, özet fonksiyonlar konularında bilgiler sunulmuş ve benzer çalışmalara yer verilmiştir. Bölüm 3'te geliştirilen algoritmanın yapısı, şifreleme adımları, her adımda kullanılan yöntemlerin detayları, kullanılan kaotik haritalar, özet fonksiyonlar, matematiksel modeller ve akış diyagramları sunulmuştur. Aynı şekilde şifrelenmiş veriden orijinal veriye ulaşmayı ifade eden şifre çözme algoritması ve yapısı detaylı bir şekilde sunulmuştur. Bölüm 4'te bir şifreleme algoritması için gerekli olan başarı kriterleri belirtilmiş ve her bir test ve analiz adımı uygulanmış, sonuçlar farklı örneklerle detaylı bir şekilde sunulmuştur. Bölüm 5'te ise geliştirmiş olduğumuz algoritmanın mevcut bulunan çalışmalar ile karşılaştırmaları yapılmış ve sonuçlar üzerinden tartışmalar yapılmıştır. Son bölüm olan Bölüm 6'da ise tez çalışması kapsamında elde edilen sonuçlar genel hatları ile değerlendirilmiş, öneriler ve gelecek çalışmalar belirtilmiştir.

2. KAVRAMSAL ÇERÇEVE

Bu tez çalışması kapsamında kaynak kısıtlı cihazlar için özet kod kullanılarak kaos tabanlı hafif sıklet bir şifreleme algoritması geliştirilmiş ve resim dosyaları üzerinde uygulanmıştır. Çalışma kapsamında kullanılan kavramların daha iyi anlaşılabilmesi için bu bölümde kavramların teorik açıklamaları yapılmış ve bu alanda yapılan çalışmalar ana hatları ile incelenmiştir. Bölüm 2.1'de kriptoloji hakkında genel bilgiler sunulmuştur. Bölüm 2.2'de hafif sıklet simetrik şifreleme algoritmaları incelenmiştir. Bölüm 2.3'te tez çalışmasının ana parçalarından biri olan kaotik sistemler incelenmiş, kaotik haritalar hakkında bilgiler sunulmuştur. Bölüm 2.4'te yine tez çalışmasında kullanılan özet fonksiyonlar ele alınmıştır. Bölüm 2.5' de ise benzer çalışmalara yer verilmiştir.

2.1. KRİPTOLOJİ

Kısaca şifre bilimi olarak tanımlayabileceğimiz kriptoloji, verinin şifrelenmesi (encryption) ve şifreli veriden orijinal verinin elde edilmesi (decryption) teknikleri ile ilgilenen kriptografi ve şifreli veriden orijinal veriyi anahtarsız elde etme (şifre kırma) teknikleri ile uğraşan kriptanaliz olmak üzere iki alt disiplinden oluşur. Şekil 2.1'de kriptoloji ve alt disiplinleri gösterilmiştir.



Şekil 2.1: Kriptoloji ve alt disiplinleri [1].

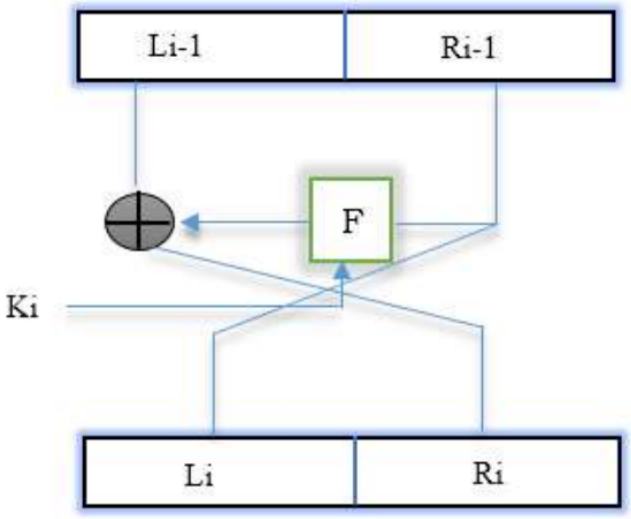
2.1.1. Kriptografi

Yunancada “gizli” anlamına gelen “kriptos” kelimesi ile “yazı” anlamına gelen “graphi” kelimelerinden türetilmiş olan kriptografi verinin yetkisiz kişiler ya da sistemler tarafından anlaşılamayacak bir forma dönüştürülmesi ve ihtiyaç halinde yetkili kişi ve sistemeler tarafında tekrar orijinal haline dönüştürülmesi teknikleri ile uğraşan bir disiplindir.

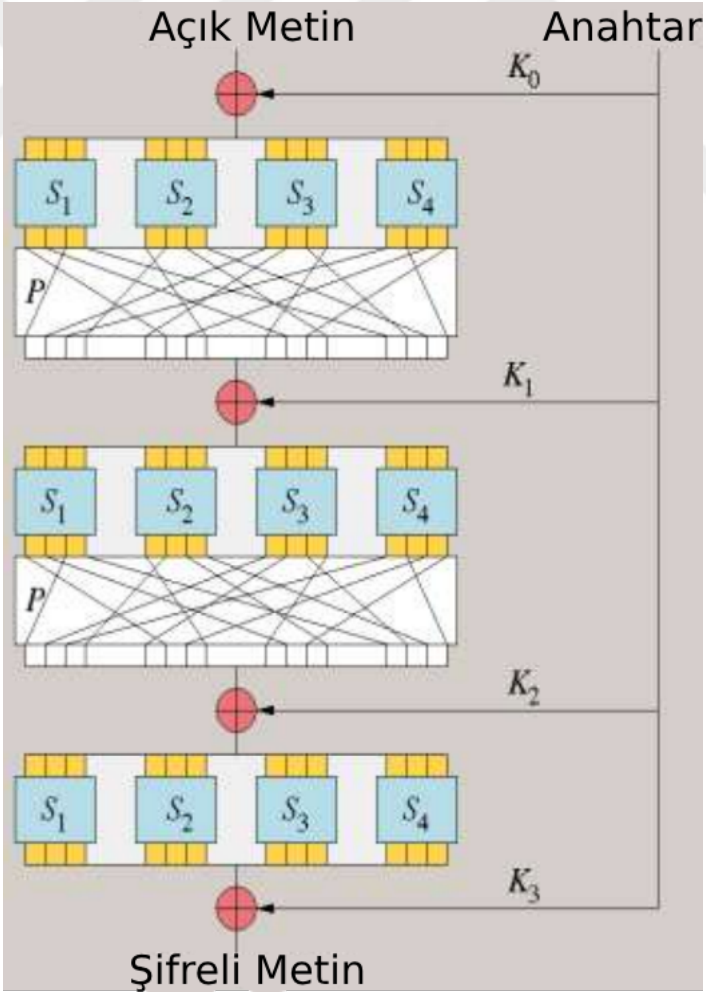
Claude Shannon’ın 1945’de yayınladığı, kriptografinin temellerinin atıldığı ve hala geçerliliğini koruyan “A Mathematical Theory of Cryptography” [26] raporuna ve 1949’da yayınlanan “Communication theory of secrecy systems” [27] makalesine göre başarılı bir şifreleme algoritması iki temel görevi başarmak zorundadır; şifrelemede kullanılan anahtar (key) ile şifrelenmiş veri (chipertext) arasındaki ilişkiyi bozmalı (confusion - karıştırma) ve açık veri (plaintext) ile şifreli veri arasındaki ilişkiyi kırmalı (diffusion - yayılma). Karıştırma için bazen confusion yerine substitution, yayılma yani diffusion yerine ise transposition ya da permutation kavramları kullanılmaktadır. Bazı şifreleme algoritmaları yalnızca karıştırma tekniklerini, bazı şifreleme algoritmaları yalnızca dağılma tekniklerini kullansa da makul bir şifreleme algoritmasının her iki tekniği birlikte kullanması beklenir.

Şifreleme algoritmaları yapıları itibariyle genel olarak ikiye ayrılı, Feistel Ağlar (Feistel Networks - FN) ve Yerine koyma – Yer değiştirme Ağları (Substitution-Permutation Networks - SPN). Shannon’ın çalışma ve önerilerine göre IBM firmasında araştırmacı olarak çalışan Horst Feistel simetrik şifreleme algoritmaları için genel bir yöntem önermiştir ve bu yöntem Feistel Ağlar olarak isimlendirilmiştir [28]. Günümüzdeki birçok şifreleme algoritmaları bu yapıyı baz almaktadır. Şekil 2.2’de Feistel ağının akışı verilmiştir.

Yerine koyma – Yer değiştirme Ağlarında ise birbirine bağlı bir dizi matematiksel işlem art arda uygulanmaktadır. Blok tabanlı şifreleme algoritmalarında genellikle bu yapı kullanılır ve en bilindik örnekleri AES, PRESENT, Square algoritmalarıdır. Şekil 2.3’de SPN akışı gösterilmiştir.



Şekil 2.2: Bir döngülü Fiestel ağı [28].



Şekil 2.3: Yerine koyma – yer değiştirme ağları (SPN) [28].

Kriptografi, gizli anahtarın kullanım şekillerine bağlı olarak, simetrik şifreleme algoritmaları, asimetrik şifreleme algoritmaları ve kriptografik protokoller olmak üzere 3 alt dala ayrılmaktadır.

2.1.1.1. Simetrik Şifreleme Algoritmaları

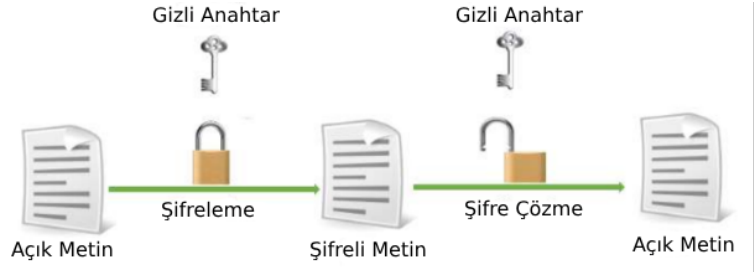
Verinin aynı anahtar ile şifrelenip çözüldüğü şifreleme algoritmalarına simetrik şifreleme algoritmaları denir. En yaygın kullanılan şifreleme tekniğidir ve yakın zaman kadar şifreleme denildiğinde akla simetrik şifreleme algoritmaları gelmekte idi. Şifrelemede kullanılan anahtar kritik öneme sahip olduğundan dolayı bazı kaynaklarda gizli anahtar (private-key) şifreleme algoritmaları olarak da geçmektedirler. Burada anahtar kritik öneme sahiptir, yeterince uzun anahtar sunmayan algoritmalar özellikle kaba saldırılara (brute force attack) karşı dayanıksızdır. Asimetrik şifreleme algoritmalarına göre daha hızlı çalışırlar.

Günümüzde yaygın olarak kullanılan bazı simetrik şifreleme algoritmalarına şunlardır;

- AES
- DES
- 3DES/TDES
- Blowfish
- RC4, RC5

Simetrik şifreleme algoritmaları veriyi işleme şekillerine göre blok şifreleme algoritmaları (block cipher algorithms) ve dizi şifreleme algoritmaları (stream cipher algorithms) diye ikiye ayrılırlar. Blok şifreleme algoritmaları veriyi bloklar/matris şeklinde ele alarak şifreleme işlemini yaparken dizi şifreleme algoritmaları veriyi bit dizisi olarak ele alır ve şifreleme işlemini gerçekleştirir.

Şekil 2.4'de simetrik şifreleme algoritmalarının gelen yapısı sunulmuştur. Şekilden de görülebileceği gibi şifreleme ve şifre çözme aşamalarında aynı anahtar kullanılmaktadır.



Şekil 2.4: Simetrik şifreleme algoritması [28].

2.1.1.2. Asimetrik Şifreleme Algoritmaları

Şifreleme aşamasında kullanılan anahtar ile şifre çözme aşamasında kullanılan anahtarların birbirlerinden farklı olduğu şifreleme algoritmalarına asimetrik şifreleme algoritmaları denir. Şifreleme aşamasında kullanılan açık (public) anahtar ile şifre çözme aşamasında kullanılan gizli (private) anahtar arasında matematiksel bir ilişki vardır fakat bu ilişkinin çözülmesi yani açık anahtardan gizli anahtarın elde edilmesi imkansıza yakın derece zor ve karmaşıktır. Bazı kaynaklarda açık anahtar (public-key) şifreleme algoritmaları olarak da isimlendirilirler. Ağır matematiksel işlemler kullanıldığından simetrik şifreleme algoritmalarına göre daha yavaş çalışmaktadırlar. Genelde kısa mesajların, örneğin simetrik şifreleme algoritmasında kullanılan gizli anahtarın, şifrelenerek paylaşılmasında kullanılırlar.

Günümüzde yaygın olarak kullanılan bazı asimetrik şifreleme algoritmaları şunlardır;

- RSA (Rivest–Shamir–Adleman)
- DSA (Digital Signature Algorithm – Dijital İmza Algoritması)
- ECC (Elliptic-curve cryptography – Eliptik Eğri Kriptografi)

Şekil 2.5'de asimetrik şifreleme algoritmalarının genel yapısı sunulmuştur. Şekilden de görülebileceği gibi şifreleme ve şifre çözme aşamalarında farklı anahtarlar kullanılmaktadır.

Şekil 2.5: Asimetrik şifreleme algoritması [28].

2.1.1.3. Kriptografik Protokoller

Simetrik ve asimetrik algoritmaların birlikte kullanılması ile daha güvenli ve hızlı bir şifreleme altyapısının sunulduğu özel şifreleme uygulamalarıdır. En yayın kullanılanı bütün internet tarayıcılarının (web browser) kullandığı Taşıma Katman Güvenliği (Transport Layer Security - TLS) protokolüdür. Burada asıl veri simetrik şifreleme algoritmaları ile şifrelenirken paylaşılan anahtar asimetrik şifreleme algoritmaları ile şifrelenir. Bu sayede hızlı ve güvenli bir iletişim sağlanmış olur.

2.2. HAFİF SIKLET ŞİFRELEME ALGORİTMALARI

Teknolojinin hızla gelişmesinin bir sonucu olarak dijital veri hayatımızın her alanına girmiştir. Giyilebilir cihazlar (wearable devices), nesnelerin interneti (Internet of things – IoT), kablosuz sensör ağlar (wireless sensor networks), RFID etiketler (RFID tag) vb. Bu tür cihazlar yapısal olarak daha düşük donanıma sahiptirler, kullanılan işlemci (Central Processing Unit - CPU), bellek (Random Access Memory - RAM), bant genişliği (bandwidth) masaüstü bilgisayarlar ya da sunucu makinalara göre oldukça düşüktür. Ayrıca bu tür cihazların çoğu doğrudan enerji kaynağına bağlı olmayıp bir batarya üzerinden enerji gereksinimlerini karşılamaktadırlar. Bu durum enerji tüketim kısıtını beraberinde getirmektedir.

Tablo 2.1: Hafif sıklet şifreleme algoritmalarından bazıları [22].

Algoritma	**Döngü sayısı**	**Anahtar Uzunluğu**	**Blok Uzunluğu**	**Yapı**
TEA [30]	64	128	64	FN
XTEA [31]	64	128	64	FN
HIGHT [32]	32	128	64	GFS
FeW [33]	32	80/128	64	FN-M
Simon [34]	32/36/42/44/52/54 /68/69/72	64/72/96/128/144/192/256	32/48/64/92/128	FNI
Speck [34]	22/23/26/27/28/29 /32/33/34	64/72/96/128/144/192/256	32/48/64/92/128	FN
PRESENT [35]	31	80/128	64	SPN
RECTANGLE [36]	25	80/120	64	SPN
LEA [37]	24/28/32	128/192/256	128	FN
Prince [38]	11	128	64	SPN
AES	10/12/14	128/192/256	128	SPN
RC5 [39]	12	128	32/64/128	FN

Günümüz kabul görmüş şifreleme algoritmaları, kompleks yapıları gereği kaynak tüketimi oldukça yüksektir ve bu tür sistemler için uygun değildir [29]. Bu durum hafif sıklet şifreleme algoritmalarının (Lightweight Crptography - LWC) geliştirilmesine yol açmıştır. Hafif sıklet şifreleme algoritmaları kısıtlı kaynak ve enerji ortamlarında gerekli güvenliği sağlayabilecek şekilde tasarlanmışlardır.

Tablo 2.1'de günümüzde yayın olarak kullanılan bazı hafif sıklet şifreleme algoritmaları ve bu algoritmaların teknik özellikleri verilmiştir. Döngü sayıları, anahtar uzunlukları, blok uzunlukları ve algoritmanın temel alındığı yapı karşılaştırılmalı olarak sunulmuştur. Gelişen teknolojiye paralel olarak özellikle kaba saldırılara karşı korunabilmek için anahtar uzunluğunun asgari 100 bit olması gerekmektedir. Bu nedenden dolayı birçok algoritma anahtar uzunluklarını gelişen teknolojiye paralel bir şekilde yenilemek zorunda kalmıştır.

2.3. KAOTİK SİSTEMLER

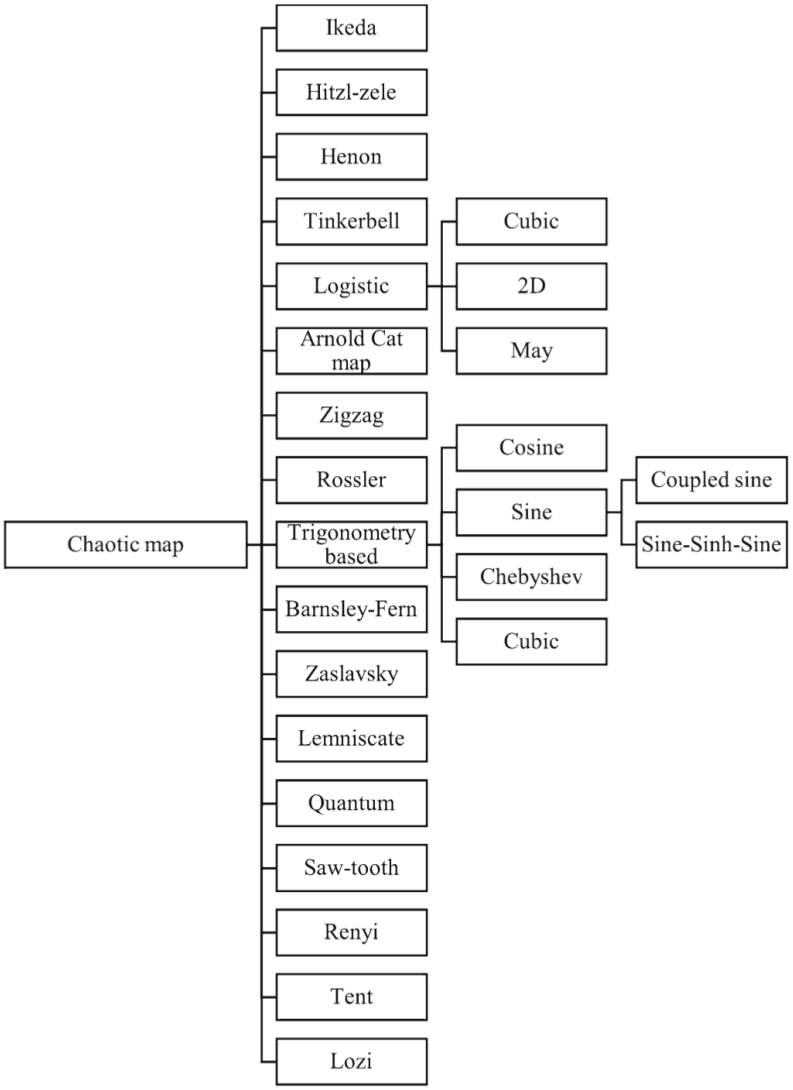
Kaos doğada birçok alanda karşımıza çıkan ve çok ilginç özellikleri bünyesinde barındıran bir fenomendir. Kaos teorisi, kaos fenomenini açıklamayı ve kaotik dinamik sistemleri kontrol ederek kaotik sistemin özelliklerinden faydalanmayı amaçlamış bir teoridir. Günümüzde kaos fizikten kimyaya, psikolojiden mühendisliğe kadar birçok alanda kullanılmaktadır [23].

Kaotik sistemler şifreme için kullanabileceğimiz birçok özelliği bünyesinde barındırmaktadır, bu özellikler şu şekilde sıralanabilir;

- İlk durum parametrelerine hassas bağlılık (Sensitive to initial conditions)
- Tekrarlanmayan çıktı (Non-periodicity)
- Çıktının belirli bir aralıkta olması (Bounded)
- Doğrusal olmama (Non-linear)
- Deterministik (Deterministic)

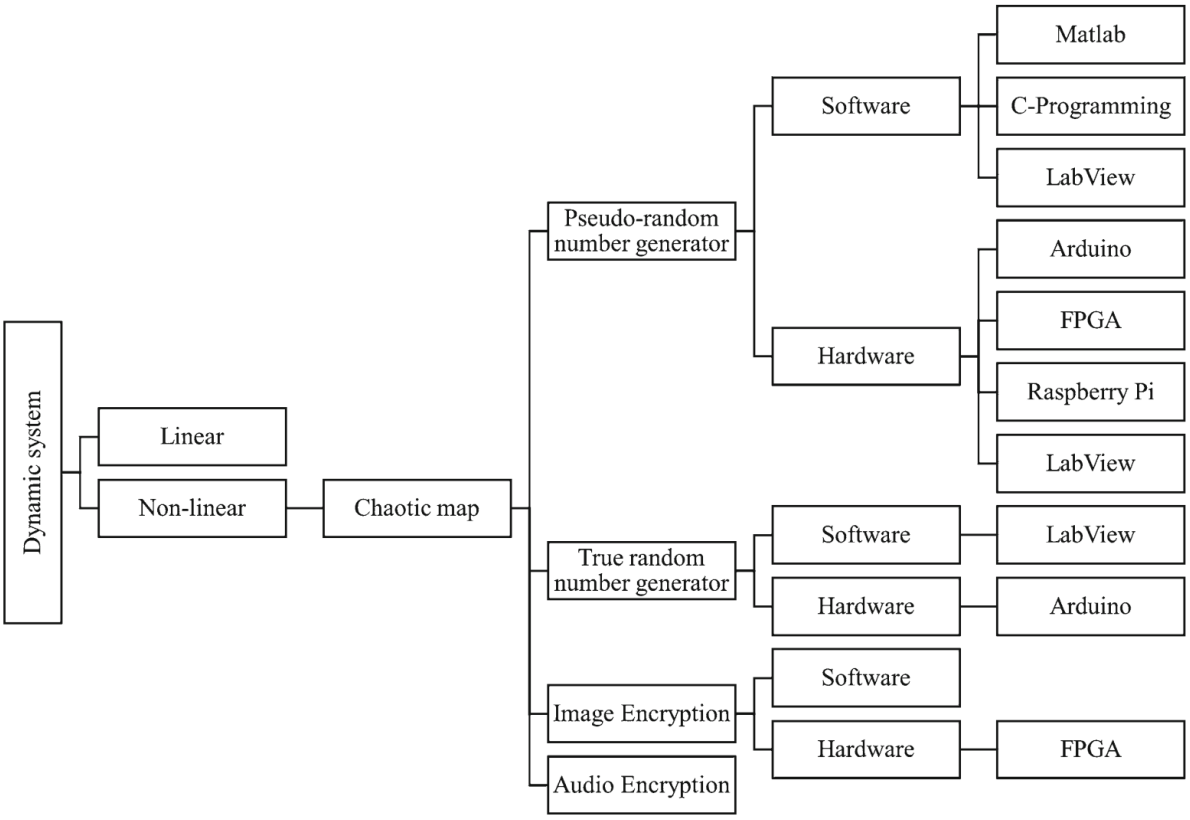
Kaotik sistemlerin bünyelerinde bulunan bu temel özellikler araştırmacıları cezbetmiş ve kriptolojide bu sistemleri kullanmaya sevk etmiştir. Özellikle son birkaç on yılda şifreleme algoritmaların tasarlanmasında yayın bir şekilde kullanılmaktadırlar [40-42]. Son zamanlarda

yapılan birçok çalışmada SPN tabanlı şifreleme algoritmalarının temelini oluşturan S-Box tasarımında da kaotik sistemlerden yararlanılmıştır [43]. Günümüzde birçok farklı özelliklere sahip kaotik harita geliştirilmiştir. Şekil 2.6'de en yaygın kullanılan kaotik haritalar listelenmiştir.



Şekil 2.6: Yaygın kullanılan kaotik haritalar [44].

Mevcut çalışmalara baktığımızda kaotik haritaların birçok alanda kullanıldıklarını görmekteyiz. Sözde rastgele sayı üreteci (Pseudo-Random Number Generator - PRNG), gerçek rastgele sayı üreteci (True Random Number Generator - TRNG), resim şifreleme (Image Encryption), video şifreleme (Video Encryption), ses şifreleme (Audio Encryption) kaotik haritaların yoğun olarak kullanıldığı alanlardan bazılarıdır [44]. Şekil 2.7'de kaotik haritaların kullanım alanları, kullanım şekilleri ve kullanılan platform ve diller gösterilmiştir.



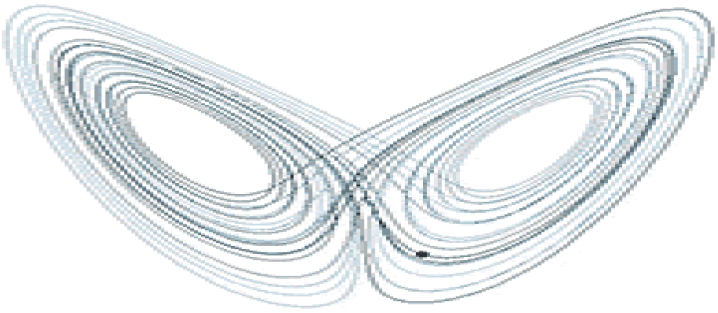
Şekil 2.7: Kaotik haritaların kullanım alanları [44].

2.3.1. Kelebek Etkisi (Butterfly Effect)

"Amazon ormanlarındaki bir kelebeğin kanat çırpması dünyayı dolaşacak bir kasırganın doğmasına neden olabilir." – Edward N. Lorenz

"Bir çivi bir nalı, bir nal bir atı, bir at bir komutanı, bir komutan bir orduyu, bir ordu koca bir ülkeyi kurtarır, o yüzden bir çiviyi asla küçümseme." – Cengiz Han

Yukarıda alıntıladığımız bu iki özdeyiş de aslında aynı konuya dikkat çekmektedir, basit, küçük zannettiğimiz olayların/olguların büyük ve beklenmedik sonuçları olabilmektedir. İşte bu etkiye kelebek etkisi (butterfly effect) denilmektedir [45]. Kaos teorisinin temelini oluşturan bu yaklaşımda, sistem girdilerinde yapılacak çok küçük ve önemsiz görüler değişiklikler çıktı üzerinde çok büyük ve tahmin edilemez etkiler oluşturabilmektedir. Kaotik sistemlerde bulunun bu özellik şifreleme algoritmaları için çok önemli ve güçlü bir altyapı sunmaktadır. Bu durum özellikle son zamanlarda kaotik sistemlerin kriptografide yoğun bir şekilde kullanılmasının önünü açmıştır.



Şekil 2.8: Kelebek etkisi [45].

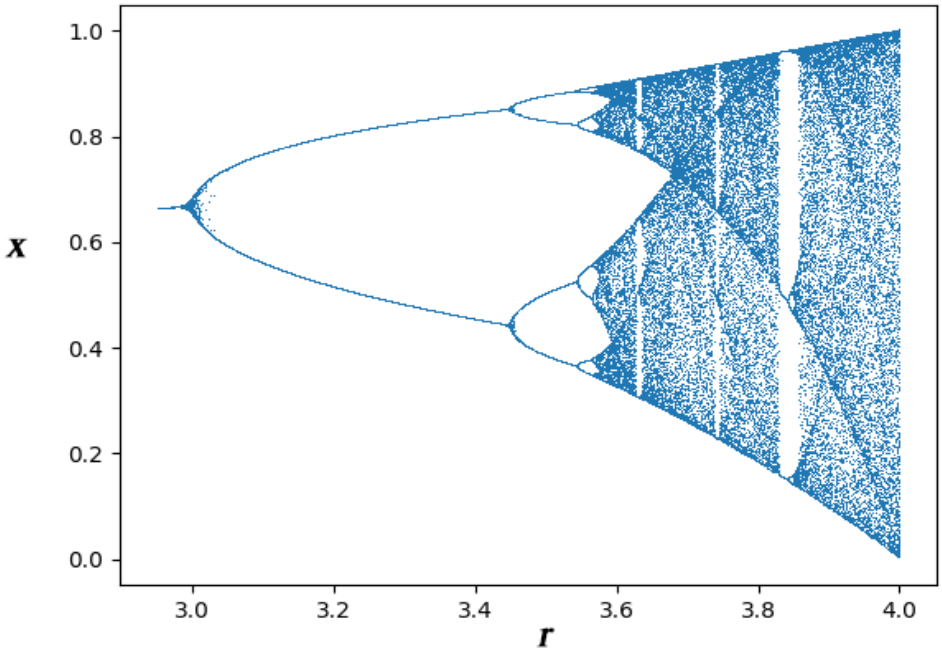
2.3.2. Lojistik Harita (Logistic Map)

Tez kapsamında geliştirilen şifreleme algoritmasında lojistik harita kullanılmıştır. Yaygın olarak kullanılan lojistik harita matematikçi Pierre François Verhulst tarafından geliştirilmiştir ve biyolog Rubert May tarafından 1976 yılında yayınlanan makalesi ile popülerleştirilmiştir. Lojistik haritanın formülü Formül (2.1)'de verilmiştir.

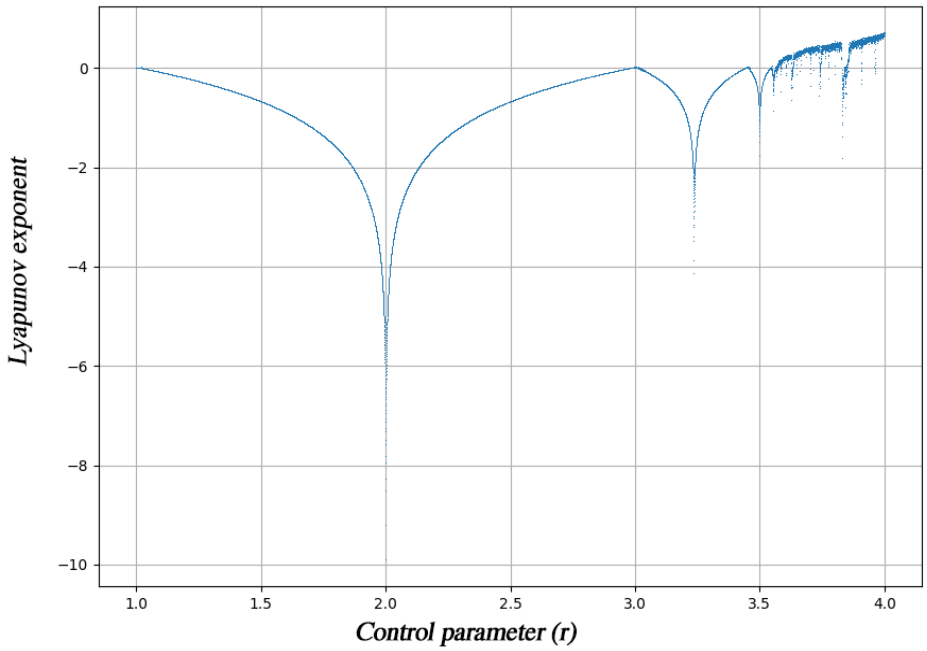
$$x_{n+1} = r x_n \ (1 - \ x_n) \tag{2.1}$$

Burada r kontrol parametresidir ve [0,4] aralığındadır, $x_n \ \in \ [0, 1]$ aralığıdan bir değer almalıdır. Sistem r değerinin $[3.5, 4]$ aralığında olduğu durumlarda kaotik özellik göstermektedir. Lojistik harita birçok şifreleme algoritmasında yayın bir kullanıma sahiptir. Basit ve sade yapısı, kolay uygulanabilirliği ve ürettiği başarılı rastgele değerler araştırmacılar tarafından bu haritanın tercih edilmesinde önemli faktörlerden olmuştur. Şekil 2.9'de lojistik harita çatallanma diyagramı gösterilmiştir.

Bir kaotik harita her durumda kaotik özellikler göstermez. Hangi durumlarda kaotik bölgeye girdiğini ölçmek için Lyapunov exponent değerine bakılır. Lyapunov exponent değeri 1'den küçük ise sistem kaotik bölgede değildir, 0 ise bir çatallanma durumu söz konusudur ve 1'den büyük ise kaotik bölgededir. Şekil 2.10'da lojistik harita Lyapunov exponent diyagramı verilmiştir.



Şekil 2.9: Lojistik harita çatallanma diyagramı.



Şekil 2.10: Lojistik harita Lyapunov exponent diyagramı.

2.4. ÖZET FONKSİYONLAR (HASH FUNCTIONS)

Özet fonksiyonlar, girdi olarak kullanılan veriden, girdi veriyi tanımlayıcı sabit uzunlukta kısa değerler üretmek için kullanılırlar. Bu tür fonksiyonlar girdi değerine karşı çok hassas olup girdi veri de yapılacak bir 1 bitlik bir değişim bile çok farklı bir özet verinin oluşmasına neden olur. Özet fonksiyonların ürettiği değerlere özet kod (hash code) ya da özet değer (hash value) denir. Her biri farklı özelliklerde ama aynı amaca hizmet eden birçok özet fonksiyon vardır. Bunlardan en yaygın olanları;

- SHA (Secure Hash Algorithm – Güvenli Özet Algoritması)
- CRC (Cyclic Redundancy Check – Döngüsel Artıklık Kontrolü)
- MurmurHash
- BLAKE2
- Argon2

Bu çalışma kapsamında bir SHA algoritması türevi olan SHA-224 özet fonksiyonu kullanılmıştır. SHA-224 fonksiyonu adından da anlaşılacağı üzere girdi verinin 224 bitlik bir özet değerini oluşturur.

2.5. ŞİFRELEME ALGORİTMASI ANALİZLERİ

Şifreleme algoritmalarının performanslarını ölçmek için çeşitli analizler yapılmaktadır. Her bir analiz algoritmanın farklı bir zafiyete karşı dayanıklılığını ölçmek için yapılmaktadır ve ulaşılması gereken minimum değer hedef olarak belirlenmiştir. Bir şifreleme algoritmasının başarılı sayılabilmesi için bütün bu analiz aşamalarından başarılı bir şekilde geçmesi gerekmektedir.

Şifreleme algoritmalarının analizleri özellikle resim dosyaları üzerinden yapılmaktadır. Bunun en büyük nedeni resim dosyalarındaki komşu pikseller arası sıkı ilişkidir. Burada 3 çeşit komşuluktan söze etmek mümkündür, yatay piksel komşuluk (horizontal pixel neighborhood), dikey piksel komşuluk (vertical pixel neighborhood) ve çapraz piksel komşuluk (diagonal pixel neighborhood). Bir resim dosyasında komşu iki piksel arasında renk geçişi genellikle sıfır ya da sıfıra çok yakın bir değerdedir. Resim dosyalarındaki bu özellik şifreleme algoritmaları için

oldukça zorlayıcı bir durum oluşturur. Başarılı bir şifreleme algoritmasının bütün bu komşuluk ilişkilerini tamamen bozması, saldırganlara (kriptanalistler) orijinal veri hakkında hiçbir ipucu vermemesi beklenir.

Geliştirmiş olduğumuz şifreleme algoritması, alandaki benzer çalışmalarda yaygın olarak kullanılan 256x256 boyutlarında 8 adet gri tonlamalı resim dosyaları üzerinden aşağıda belirtilen test ve analizlere tabi tutulmuştur. Her bir test ve analizi gerçekleştirmek için python programlama dilinde özel senaryolar geliştirilmiş, çıktılar farklı formatlarda dosyalara kaydedilmiştir.

- Anahtar Analizi (Key Analysis)
    - Anahtar Uzayı Analizi (Key Space Analysis)
    - Anahtar Hassasiyet Analizi (Key Sensitivity Analysis)
- Bilgi Entropi Analizi (Information Entropy Analysis)
- İstatistiksel Saldırı Analizi (Statistical Attack Analysis)
    - Histogram Analizi (Histogram Analysis)
    - Korelasyon Analizi (Correlation Analysis)
- Diferansiyel Saldırı Analizi (Differential Attack Analysis)
- Dayanıklılık Analizi (Robustness Analysis)
    - Kırpma Saldırısı (Cropping attack)
    - Gürültü Saldırısı (Noise attack)
    - Bilinen Veri ve Seçilen Veri Saldırısı (Known-plaintext and Chosen-plaintext attacks)
- Güvenlik ve Performans Analizi (Security and Performance Analysis)
    - Güvenlik Analizi (Security Analysis)

- Zaman Karmaşıklık Analizi (Time Complexity Analysis)

2.6. MEVCUT ÇALIŞMALAR

Konu ile ilgili mevcut çalışmalara bakıldığında, tez çalışmasına konu olan kaos tabanlı hafif sıklet şifreleme algoritması alanında özellikle son zamanlarda birçok çalışma yapıldığını görmekteyiz. Araştırmacılar farklı farklı teknikler kullanarak algoritmanın kaynak tüketimini en aza indirirken veri güvenliğini maksimize etmeyi hedeflemişlerdir.

Alawida ve diğ. [46], kaotik sistemlerdeki kompleksliği arttırmak, daha geniş kaotik alan oluşturmak, ilk durum parametrelerine olan bağlılığı daha fazla hassaslaştırmak, çıktının doğrusallığını daha fazla bozmak için kosinüs fonksiyonundan yararlanmışlardır. Önerdikleri yöntemi lojistik harita ve henon haritalara uygulamışlar ve bu kaotik haritaların tek başlarına uygulanması ile elde ettikleri sonuçlardan çok daha uygun sonuçlar elde etmişlerdir. Geliştirdikleri yeni modeli sahte rastgele sayı üretici olarak uyarlamışlar ve başarılı sonuçlar elde etmişlerdir.

Alibrahim ve diğ. [47], lojistik harita ve parçacık sürü optimizasyonu (Particle Swarm Optimization - PSO) algoritmasını birlikte kullanarak yeni bir resim şifleme algoritması geliştirmişlerdir. Bu yeni geliştirilen algoritmada lojistik harita karıştırma ve yayılma işleminde kullanılırken PSO pikseller arasındaki korelasyonun hesaplanması ve en düşük korelasyona sahip piksellerin tespit edilmesi ve seçilmesi adımlarında kullanılmıştır. Geliştirmiş oldukları algoritmanın resim dosyaları üzerinden yapılan test ve analiz sonuçları mevcut resim şifreleme algoritmalarına göre daha başarılı sonuçlar üretmiştir.

Arab ve diğ. [48], kaotik sistemleri kullanarak AES algoritmasında değişiklikler yapmış ve yeni bir resim şifreleme algoritması geliştirmişlerdir. Şifreleme anahtarı Arnold haritası kullanılarak üretilmiş, şifreleme işlemi değiştirilmiş AES algoritması ile yapılmış ve her bir döngüde (round) ihtiyaç duyulan anahtar yine Arnold haritası kullanılarak üretilmiştir. Yeni geliştirilen algoritmada zaman kompleksliği düşmüştür. Diferansiyel saldırılarına ve kaba kuvvet saldırılarına karşı algoritma daha dayanıklıdır. Bilgi entropisi ideal değerlere çok yakın çıkmıştır. İlk durum parametrelerine daha hassas bir bağlılık elde edilmiş olup parametrelerde yapılan çok ufak değişiklikler çok farklı sonuçların elde edilmesine neden olmuştur.

Babaei ve diğ. [49], lojistik harita ile DNA (Deoxyribonucleic acid - Deoksiriboz nükleik asit) modelini birlikte kullanarak yeni bir şifreleme algoritması geliştirmişlerdir. Bu algoritma daha çok resim şifreleme ve metin şifreleme amacı ile geliştirilmiştir. Kodlama ve simülasyonlar Matlap üzerinde yapılmış ve sonuçlar AES algoritması ile karşılaştırılmıştır ve algoritmanın oldukça başarılı çıktılar ürettiği gözlemlenmiştir.

Bakhshandeh ve diğ. [50], kaotik haritalar ve hücresel otomatlar kullanarak yeni bir resim şifreleme algoritması geliştirmişlerdir. Karıştırma ve yayılma mimarisi kullandıkları bu yeni şifreleme algoritmasında, karıştırma aşamasında parçalı ve lineer kaotik harita kullanırken yayım aşamasında lojistik harita ve hücresel otomatlar birlikte kullanılmıştır. Bu sayede daha efektif bir şifreleme algoritması geliştirilmiştir. Geliştirilen algoritmanın yüksek güvenlik, düşük hesaplama ve kolay uygulanabilirlik gibi özelliklerinin yanında önemli bir özelliği daha vardır, dosyanın gönderimi esnasında bir bozulma olup olmadığının tespit edilebilir olmasıdır. Bu özellik mevcut şifreleme algoritmalarına göre yeni geliştirilen algoritmayı üstün kılmaktadır ve pratik bir resim şifreleme algoritması olarak değerlendirilmesini arttırmaktadır.

Bhattacharjee ve diğ. [51], COVID-19 medikal imajlarının güvenliği için kaos tabanlı hafif sıklet bir şifreleme algoritması geliştirmişlerdir. Geliştirilen algoritma üç aşamadan oluşmaktadır; ilk adımda değişken uzunluktaki gri tonlamalı resim dosyası kullanılarak bir gizli anahtar üretilmiştir. Bu anahtar daha sonra kaotik haritanın başlangıç parametresi olarak kullanılacaktır. İkinci aşamada resim dosyasının pikselleri ilk kaotik haritadan üretilen veriler doğrultusunda karıştırılmıştır. Son aşamada ise iki kaotik harita kullanılarak elde edilen veriler ile bütün resim dosyası üzerinde yayılma işlemi uygulanmıştır ki bu sayede istatistiksel ve diferansiyel saldırılarına karşı dirençli bir algoritma geliştirilmiştir. Geliştirilen algoritma bütün test ve analiz adımlarına tabi tutulmuş ve başarılı sonuçlar elde edilmiştir.

Chai ve diğ. [52], DNA modelini ve kaotik sistemleri kullanarak bir resim şifreleme algoritması geliştirmiştir. İlk aşamada açık resim dosyası DNA matrisine dönüştürülmüştür. Daha sonra dalga tabanlı bir yer değiştirme işlemi uygulanmıştır. İki boyutlu lojistik harita kullanılarak elde edilen veriler ile satır tabanlı dairesel yer değiştirme (row circular permutation - RCP) ve sütün tabanlı dairesel yer değiştirme (column circular permutation – CCP) işlemleri uygulanmıştır. Şifrelenecek resim dosyasının SHA 256 özet fonksiyonu kullanılarak özet kodu elde edilip kaotik haritanın giriş parametresi olarak kullanılmıştır. Daha sonra satır bazlı DNA operasyonu uygulanmıştır. Sistem başlangıç parametreleri ve orijinal resim dosyasının

hamming mesafesi (hamming distance) kullanılarak yenilenmiş ve nihai şifrelenmiş resim dosyası elde edilmiştir. Yapılan test ve analiz sonuçları geliştirilen algoritmanın saldırılara karşı güvenli olduğunu göstermiştir.

Chai ve diğ. [53], renkli resim dosyaları için dinamik DNA modelini ve kaotik sistemleri kullanarak yeni bir şifreleme algoritması geliştirmişlerdir. İlk olarak renkli (RGB – Red Green Blue – Kırmızı Yeşil Mavi) resim dosyası kırmızı (red), yeşil (green) ve mavi (blue) katmanlara ayrıştırılmıştır. Daha sonra bu komponentler karıştırılmıştır. Karıştırılan bu katmanlar birleştirilerek DNA matrisine dönüştürülmüştür. Kaotik haritadan elde edilen rastgele sayılar ile karıştırma işlemi uygulanmıştır. Karıştırılan bu DNA matrisine DNA çözümleme kuralları uygulanmış ve bütün veri üç ayrı eşit dosyalara bölünmüştür. Son aşamada algoritmanın güvenliğini arttırmak için tekrar bir karıştırma işlemi uygulanmıştır. Bu geliştirilen şifreleme algoritmasında, sözde rastgele sayı üreteci olarak dört kanatlı aşırı kaotik harita kullanılmıştır. Orijinal resim dosyasının SHA-384 özet fonksiyonu kullanılarak özet değeri elde edilmiş ve bu değer diğer parametrelerle birleştirilerek kaotik haritanın başlangıç durum parametrelerinin belirlenmesinde kullanılmıştır.

Dhall ve diğ. [54], resim dosyalarını şifrelemek için kaos tabanlı olasılıksal yeni bir şifreleme algoritması geliştirdiler. Mevcut şifreleme algoritmaları bir dosyayı aynı anahtar ile şifrelediğinde aynı çıktıyı üretmektedirler. Bu durum bazı güvenlik zafiyetlerine neden olmaktadır. Bu noktadan hareketle aynı dosyanın aynı anahtarla şifrelenmesi durumunda bile farklı bir çıktı elde edilmesini sağlayacak yeni bir şifreleme algoritması geliştirilmiştir. Geliştirilen bu yeni algoritma blok tabanlı simetrik bir şifreleme yapısında olup ilk aşamada şifrelenecek veri içerisine rastgele veri ekleme (Random Bits Insertion) işlemi yapılmaktadır. Bu sayede aynı girdi ve anahtar için farklı çıktılar elde edilebilmiştir. Daha sonra XoR (exclusive-OR – özel OR) işlemi uygulanarak şifreleme işlemi yapılmıştır.

Fang ve diğ. [55], karşıt ağlar (Adversarial Networks – GANs) ve DNA modellerini kullanarak kaos tabanlı bir blok resim şifreleme algoritması geliştirdiler. İlk aşamada tek boyutlu kaotik sistem GANs ve DNA dizilerini kullanarak iki kompleks anahtar dizisi oluşturulmuştur. Daha sonra bu iki yeni rastgele dizilere Feistel ağları kullanılarak karıştırma ve yayılma işlemi uygulanmış ve bu sayede şifrelenmiş veri elde edilmiştir.

Faragallah [56], Henon ve Standart haritalarını kullanarak kaos tabanlı yeni bir resim şifreleme algoritması geliştirmiştir. Geliştirilen algoritma iki aşamadan oluşmaktadır. İlk

aşamada Standart harita değiştirilerek elde edilen yeni harita ile orijinal resim dosyası karıştırılmıştır. Bu karıştırma işlemi *n* kez tekrarlanmıştır. İkinci aşamada ise bu karıştırılan resim dosyasına Henon haritası uygulanmıştır. Bütün bu adımlar m kez tekrarlanarak güvenli bir şifreleme gerçekleştirilmiştir.

Farah ve diğ. [57], optik görüntüler (optical images) için kaos tabanlı, DNA ve kısmı Fourier dönüşümü (Fractional Fourier Transform) yöntemlerini kullanarak yeni bir şifreleme algoritması geliştirmişlerdir. İlk aşamada düz veri DNA matrisine dönüştürülmüş ve Lorenz haritası kullanılarak rastgele maskeleme değerleri üretilmiştir. Daha sonra bu DNA matrisi ile rastgele maskeleme değerleri birleştirilmiş ve üç kez kısmi Fourier dönüşümü uygulanmıştır ve bu sayede şifrelenmiş resim dosyası elde edilmiştir.

Gafsi ve diğ. [58], Xilinx FPGA-Zynq platformu için gerçek zamanlı (real time) donanımsal (hardware) yeni bir resim şifreleme ve çözme algoritması geliştirdiler. Bu çalışmada sahte rastgele sayı üreteci olarak kaotik sistemlerden yararlanmışlardır. Geliştirilen algoritma Vivado sistem aracılığı ile Xilinx FPGA-Zynq platformuna uyarlanmıştır.

Guo ve diğ. [59], çalışmalarında üç boyutlu döngüsel bit kaydırmalı karıştırma (3D cyclic shift bit scrambling) resim şifreleme algoritması geliştirmişlerdir. İlk aşamada resim dosyasındaki piksel değerleri bir diziye daha sonra üç boyutlu bir matrise dönüştürülmüştür. Lojistik Fibonacci (Logistic-Fibonacci) kaotik haritası kullanılarak rastgele değerler üretilmiş ve bu değerler ile şifreleme işlemi yapılmıştır. Ayrıca orijinal resim dosyasının özet kodu, SHA-256 özet fonksiyonu kullanılarak elde edilmiş ve bu değer kaotik haritanın başlangıç parametrelerinde kullanılarak orijinal resim dosyasının içeriğine karşı hassas bir şifrelenmiş dosya elde edilmiştir.

Hoang ve diğ. [60], lojistik harita kullanılarak özgün bir resim şifreleme algoritması geliştirmişlerdir. Karıştırma aşamasında her bir döngüde kontrol parametreleri bit seviyesinde değiştirilerek dinamik bir yapı kurgulanmıştır. Bu sayede özellikle istatistiksel saldırılara karşı oldukça dirençli bir şifreleme algoritması elde edilmiştir.

Khan ve diğ. [61], özellikle seçilmiş açık metin saldırılarına (Chosen-Plaintext Attack) karşı koyabilmek için yeni bir resim şifreleme algoritması geliştirdiler. Bu çalışmada öncelikle düz resim dosyası ortogonal eşleşmeli takip (Orthogonal Matching Pursuit) algoritması

kullanılarak sıkıştırılmıştır. Daha sonra sıkıştırılan bu veri TD-ERCS ve Skew-tent kaotik haritaları kullanılarak şifrelenmiştir.

Lin ve diğ. [62], video ve ses dosyalarının ağ üzerinden güvenli bir şekilde iletilebilmesi için kaos tabanlı yeni bir şifreleme algoritması geliştirmişlerdir. Gönderici ve alıcı tarafta senkronizasyonun sağlanabilmesi için dinamik kaotik rastgele sayı üreteci hem gönderici hem de alıcı tarafta eş zamanlı çalıştırılmıştır. Ayrıca entegrasyonu sağlayabilmek için SHA-256 özel fonksiyonu kullanılmıştır. AES algoritması bu kurgulanan yeni yapıya uyarlanarak hem gönderici tarafta verinin şifrelenmesi hem de alıcı tarafta şifreli bu verinin çözülmesi sağlanmıştır.

Lui ve diğ. [63], renkli resim dosyalarının şifrelenmesi için yeni bir algoritma geliştirmişlerdir. Bu geliştirilen algoritmada DNA modeli ve çift kaotik harita kullanılmıştır. İlk aşamada renkli resim dosyasının her bir katmanını karıştırmak için Arnold kaotik haritası kullanılmıştır. İkinci aşamada Lorenz kaotik haritası ve Rossler aşırı kaotik haritaları kullanılarak her bir resim katmanı için farklı diziler oluşturulmuş ve yayılma işlemi gerçekleştirilmiştir ve son aşamada DNA operasyonu uygulanarak şifreleme işlemi tamamlanmıştır.

Lui ve diğ. [64], optik görüntülerin güvenliği için aşırı kaotik haritalar kullanarak yeni bir şifreleme algoritması geliştirmiştir. Çalışmalarında iki ayrı aşırı kaotik harita kullanmışlardır, bunlar, Four-Wing ve Chen 4D haritalarıdır.

Lui ve diğ. [65], renkli resim dosyalarının şifrelenmesi için dinamik DNA modeli ve 4-D membristif aşırı kaotik haritalar kullanarak yeni bir şifreleme algoritması geliştirmişlerdir. İlk aşamada kaotik matrisler 4-D membristif aşırı kaotik harita kullanılarak oluşturulmuştur. Daha sonra renkli resim dosyasındaki her bir katman ayrı bir matrise taşınmış ve DNA operasyonları her bir katman için uygulanmıştır.

Machkour ve diğ. [66], iki boyutlu lojistik harita ve Latin karesi (Latin Square) modellerini kullanarak yeni bir resim şifreleme algoritması geliştirmişlerdir. Yaptıkları birçok analiz ve testler geliştirilen algoritmanın saldırılara karşı (istatistiksel saldırılar, kaba kuvvet saldırıları vs.) oldukça dayanıklı olduğunu göstermiştir.

Malik ve diğ. [67], aşırı kaotik haritalar ve DNA modelini kullanarak renkli resim dosyaları için yeni bir şifreleme algoritması geliştirmişlerdir. İlk aşamada Tent haritası kullanılarak farklı matrisler oluşturulmuştur. Daha sonra orijinal renkli resim dosyası üç ayrı kanala ayrılmış ve karıştırma ve yayılma işlemleri uygulanmıştır. Son aşamada ise DNA modeli uygulanarak nihai şifrelenmiş resim dosyası elde edilmiştir.

Zhao ve diğ. [68], çalışmalarında IoT cihazlar için kaos tabanlı donanımsal bir kimlik doğrulama çözümü sunmuşlardır. Koyuncu ve diğ. [69], çalışmalarında FPGA (Field-programmable Gate Array – Programlanabilir Kapı Dizileri) sistemler için kaos tabanlı gerçek sayı üreteci tasarımı ve uygulaması yapmışlardır. Tutueva ve diğ. [70], çalışmalarında sözde sayı üreteci olarak kaotik sistemlerden yararlanmışlardır. Alcin ve diğ. [71], çalışmalarında FPGA cihazlar için yapay zekâ tabanlı kaotik gerçek rastgele sayı üreteci tasarlamışlardır. Zaher ve diğ. [72], çalışmalarında kaos tabanlı güvenli bir iletişim çözümü geliştirmişlerdir. Yasser ve diğ. [73], çalışmalarında güvenli iletişim için kaos tabanlı bir şifreleme altyapısı sunmuşlardır. Wang ve diğ. [74], çalışmalarında üç boyutlu modellerin şifrelenmesi için kaos tabanlı bir çözüm sunmuşlardır. Wang ve diğ. [75], Zhao ve diğ. [76], Zhang [77] ve Çavuşoğlu ve diğ. [78] resim şifreleme çalışmalarında çeşitli kaotik sistemlerden yararlanmışlardır.

Reddy ve diğ. [79], Muhammed ve diğ. [80], Tuncer [81], Papadimitriou ve diğ. [82], Leung ve diğ. [83], Guo ve diğ. [84], Kanso ve diğ. [85], Puthuparambil ve diğ. [86], çalışmalarında kaotik sistemleri kullanarak simetrik olasılıksal şifreleme algoritmaları geliştirmişlerdir.

3. YÖNTEM

Kaynak kısıtlı platformlar için geliştirilmiş birçok hafif sıklet şifreleme algoritması bulunmaktadır. Bu algoritmalar genellikle mevcut şifreleme algoritmaların güncellenmesi, döngü sayılarının azaltılması, karıştırma ve yayılma adımlarının optimize edilmeleri şeklinde ortaya çıkmıştır. Özellikle son zamanlara hafif sıklet şifreleme algoritmalarında farklı matematiksel modellerden yararlanılmaktadır. Kaotik sistemlerin sunmuş oldukları özellikler şifreleme algoritmaları için çok uygun altyapı sağlamakta ve bu durum araştırmacıların ilgisini çekmektedir. Özellikle ilk durum parametrelerine olan sıkı bağlılık kaos tabanlı haritaların tercih edilmesinde en büyük etkeni oluşturmaktadır.

Kaotik haritaların şifreleme algoritmalarında tek başlarına kullanılması, Shannon'ın kriptografinin temellerini belirlediği kriterleri karşılamakta yetersiz kalmaktadır. Bundan dolayı araştırmacılar kaotik haritaları farklı matematiksel modeller ile birlikte kullanarak istenilen düzeyde güvenliğe ulaşmayı hedeflemişlerdir. Bu modeller arasında DNA özellikle dikkat çekmektedir [87-91]. Fourier dönüşümü [57], GANs [55], Hilbert Curves ve H-Fractals [92], Knuth–Durstenfeld [93], Transcendental Numbers [94], Particle Swarm Optimization [47], Latin Square [66], Rotor Machine [95], Discrete Wavelet Transform (DWT) [96], özet fonksiyonlar diğer modeller olarak karşımıza çıkmaktadır.

Bu çalışma kapsamında özet kod kullanarak kaos tabanlı hafif sıklet bir şifreleme algoritması geliştirilmiştir ve bu bölümde geliştirilen algoritma detaylı bir şekilde açıklanmıştır. Bu bölüm şu alt başlıkları içermektedir; Bölüm 3.1'de çalışmanın ana bileşenini oluşturan lojistik harita açıklanmıştır. Bölüm 3.2'de çalışmada kullanılan özet fonksiyon hakkında bilgi sunulmuştur. Bölüm 3.3'te şifreleme adımları detaylı bir şekilde ele alınmıştır. Bölüm 3.4'te şifre çözme adımları detaylı bir şekilde sunulmuştur.

3.1. ALGORİTMANIN GENEL YAPISI VE KULLANILAN MODELLER

Bu bölümde algoritma yapısında kullanılan modeller ve bu modellerin hangi amaçlar için kullanıldığı açıklanmıştır.

3.1.1. Rastgele Sayı Üreteci

Bu çalışma kapsamında rastgele sayısı üreteci olarak, basit yapısı, kolay uygulanabilirliği, yaygın kullanımı vs. nedenlerden dolayı iki adet bir boyutlu (1-D) lojistik harita kullanılmıştır. İlk lojistik haritanın başlangıç değerleri gizli anahtardan türetilmiştir ve buradan üretilen değerler satır bazlı rotasyon, sütun bazlı rotasyon, özet değerin şifrelenmesi (XoR işlemi) ve özet değerin resim dosyası içerisine dağıtılmasında kullanılmıştır.

İkinci lojistik haritanın başlangıç değerleri gizli anahtar ve orijinal dosyanın özet kodu kullanılarak türetilmiştir. Burada üretilen değerler resim dosyasın şifrelenmesinde (XoR işlemi) kullanılmıştır.

3.1.2. Orijinal Dosyaya Bağlılık Fonksiyonu

Orijinal dosyaya bağımlılığın sağlanabilmesi yani orijinal dosyada oluşabilecek bir bitlik değişimin çıktı üzerinde çok büyük etkiler oluşturabilmesi için orijinal dosyanın özet kodundan yararlanılmıştır. Özet fonksiyon olarak SHA2-224 fonksiyonu kullanılmıştır. Şifreleme işleminde başlamadan önce özet fonksiyon ile orijinal verinin özet kodu üretilmiş ve oluşturulan bu özet değer gizli anahtar ile birlikte ikinci lojistik haritanın başlangıç parametrelerinin belirlenmesinde kullanılmıştır. Bu sayede orijinal dosyadaki çok küçük değişiklikler ikinci kaotik haritanın başlangıç değerlerinin belirlenmesinde kullanılan özet değerin çok farklı oluşmasına ve bu durum da kaotik sisteminin çok farklı değerler üretmesine ve nihayetinde şifrelenmiş dosyanın çok büyük oranda farklılığa uğramasına sebep olmuştur.

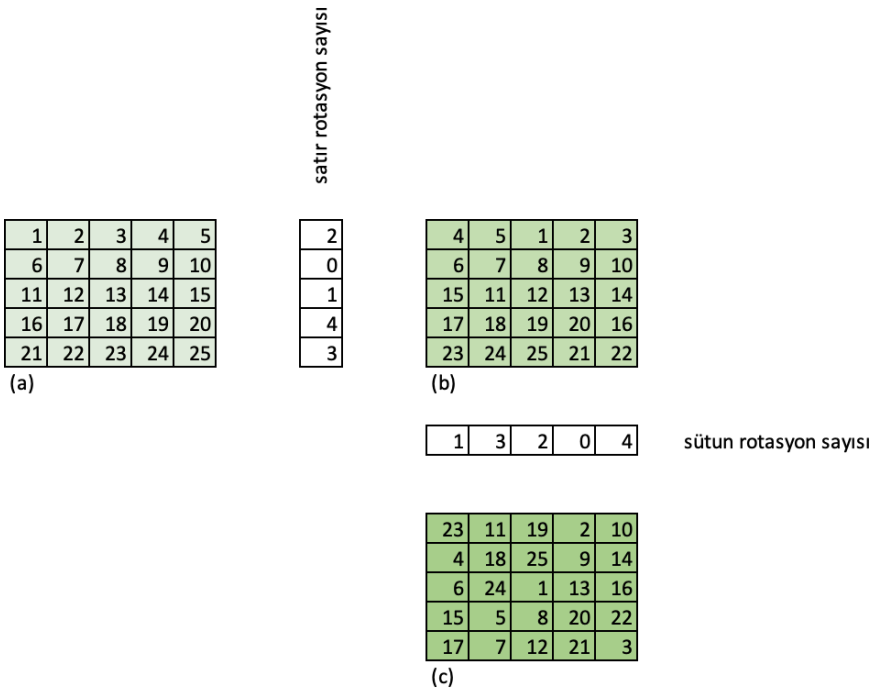
Bu durum özellikle diferansiyel saldırılara karşı şifreleme algoritmasının dirençli olabilmesi için çok önemli bir özelliktir. Başarılı bir şifreleme algoritması, orijinal dosyada ki bir bitlik değişimde bile çıktının 99% üzerinde değişime uğramasını sağlamalıdır (bu durum NPCR değeri ölçülerek belirlenir). Girdi üzerinde yapılan bir bitlik değişim özet değerin farklılaşmasına neden olmaktadır. Bu özet değeri algoritmada ikinci kaotik haritanın başlangıç parametrelerinin belirlenmesinde kullanılmaktadır ve buradaki farklılık kaotik haritanın tamamen farklı bir çıktı sunmasına sebep vermektedir. Çünkü kaotik sistemler başlangıç parametrelerine karşı çok hassas sistemlerdir. İkinci kaotik haritanın çıktıları verinin şifrelenmesinde doğrudan kullanıldığı için şifrelenmiş verinin 99% üzerinde değişmesi sağlanmış olmaktadır.

3.1.3. Karıştırma Fonksiyonları

Geliştirilen algoritmada karıştırma işlemi her bir satırın ve her bir sütunun değeri kadar rotasyona tabi tutulması ile sağlanmıştır. Birinci kaotik haritadan her bir satır için sütun uzunluğu kadar rastgele bir değer elde edilmiş ve her satır kendisi için üretilen değer kadar soldan sağa rotasyona tabi tutulmuştur. Aynı şekilde her bir sütun için satır sayısı kadar rastgele değer üretilmiş ve her bir sütun bu değerler kadar yukarıdan aşağıya rotasyona tabi tutulmuştur. Şekil 3.1'de bu durum örnek bir matris üzerinde uyarlanmıştır.

Şifre çözme adımında ise buradaki işlemler tersten yapılmaktadır. Öncelikle birinci kaotik haritadan elde edilen değerler ile sütunlar ters yönde (aşağıdan yukarıya) ve daha sonra satırlar yine ters yönde (sağdan sola) rotasyon işlemine tabi tutulmuştur. Bu şekilde komşu pikseller arasındaki bağlantılar kopartılmış, orijinal dosya ile şifrelenmiş dosya pikselleri birbirlerinden konum olarak ayrıştırılmış olmaktadır. Bu bize korelasyon analizinde yatay, dikey ve çapraz korelasyon katsayılarının çok düşük çıkması ve bu şekilde yapılacak saldırıların önlenmesini sağlamaktadır.

Ayrıca bu şekilde karıştırma işlemi kırpma saldırılarına karşı algoritmanın karşı koyabilmesi için elzemdir. Kırpma saldırılarında saldırganlar şifrelenmiş dosyanın farklı bölgelerinin tamamen siyah ya da tamamen beyaz olacak şekilde değiştirmekte ve bu şekilde farklılaştırılan dosya şifre çözücü ile açılmaktadır. Açılan dosya üzerinden orijinal veri, gizli anahtar ve şifreleme algoritmasının yapısı hakkında bilgiler sızdırılmaya çalışılmaktadır. Başarılı bir şifreleme algoritmasının kırpma işlemini orijinal dosyanın tamamına yansıtması beklenir. Ayrıca kırma işlemi ile veri kaybı olsa bile orijinal dosyanın başarılı bir şekilde edilmesi beklenmektedir.



Şekil 3.1: Döngüsel rotasyon uyarlaması.

a) orijinal matris, b) satır bazlı döngüsel rotasyon uygulandıktan sonra oluşan matris, c) sütun bazlı döngüsel rotasyon uygulandıktan sonra oluşan matris.

3.1.4. Şifreleme Fonksiyonu

Şifrele fonksiyonu olarak mantıksal işlemlerden XoR işlemi uygulanmıştır. Her bir bit XoR işlemine tabii tutularak şifreleme gerçekleştirilmiştir. XoR mantık işlemi yapısı gereği şifreleme algoritmalarında çok yaygın olarak kullanılmaktadır. İki girdinin karşılığında üretilen tek çıktı vardır ve bu çıktıdan girdilerden birinin değerini bilmeden diğer girdiyi elde etmek imkansızdır. Tablo 3.1' de XoR mantık işlemi verilmiştir.

Tablo 3.1: XoR Mantık İşlemleri

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

XoR işlemlerinin şöyle bir özelliği vardır, XoR'lanmış bir veri (tabloda Q ile belirtilen kolon) girdilerden birisi ile (A kolonu ya da B kolonu) XoR'lanacak olursa diğer kolonun değeri elde edilir (A ile XoR'lanırsa B değeri, B ile XoR'lanırsa A değeri). Bu durumda şifreleme işlemini temelin oluşturmaktadır.

Tablo 3.2'de bit bazlı şifreleme ve şifre çözme aşamaları XoR işlemi ile gösterilmiştir. Tabloda K ile belirtilen kolonlar anahtarı, P ile belirtilen kolonlar şifrelenecek açık veriyi, C ile belirtilen kolonlar ise şifrelenmiş veriyi ifade etmektedirler. Birinci satırda şifreleme aşamasında 0 değeri 0 anahtarı ile XoR'lanarak şifrelenmiş 0 şifreli verisi elde edilmiştir. Şifre çözme aşamasında da 0 şifrelenmiş verisi 0 anahtarı ile tekrar XoR'lanmış ve şifre çözülmüştür. 3. Satırda şifreleme aşamasında 1 anahtar değeri ile 0 orijinal verişi şifrelenmiş ve 1 şifrelenmiş değeri elde edilmiştir. Şifre çözme aşamasında ise 1 anahtar değeri ile 1 şifrelenmiş verisi XoR'lanarak 0 orijinal verisine ulaşılmıştır.

Tablo 3.2: XoR ile şifreleme ve şifre çözme işlemleri

Şifreleme			**Şifre Çözme**		
K	**P**	**C**	**K**	**C**	**P**
0	0	0	0	0	0
0	1	1	0	1	1
1	0	1	1	1	0
1	1	0	1	0	1

3.2. ÖNERİLEN ALGORİTMA

Önerilmen şifreleme algoritmasında iki farklı lojistik harita rastgele sayı üreteci olarak kullanılmıştır. Orijinal dosyanın özet değeri ikinci haritanın başlangıç değerlerinin belirlenmesinde bir parametre olarak kullanılmış ve bu sayede diferansiyel saldırılara karşı dirençli bir yapı oluşturulmuştur. Karıştırma işlemleri için satır ve sütun bazlı döngüsel rotasyon işlemleri uygulanmış ve gerekli döngü adım sayıları birinci kaotik haritadan elde edilmiştir. Bütün şifreleme ve şifre çözme adımları aşağıda detaylı bir şekilde açıklanmıştır.

3.2.1. Şifreleme Adımları

Şifreleme işlemine başlamadan önce şifrelenecek olan açık verinin SHA2-224 özet fonksiyonu kullanılarak özet değeri elde edilir. Bu özet kodun ilk 2 byte'ı gizli anahtar değeri

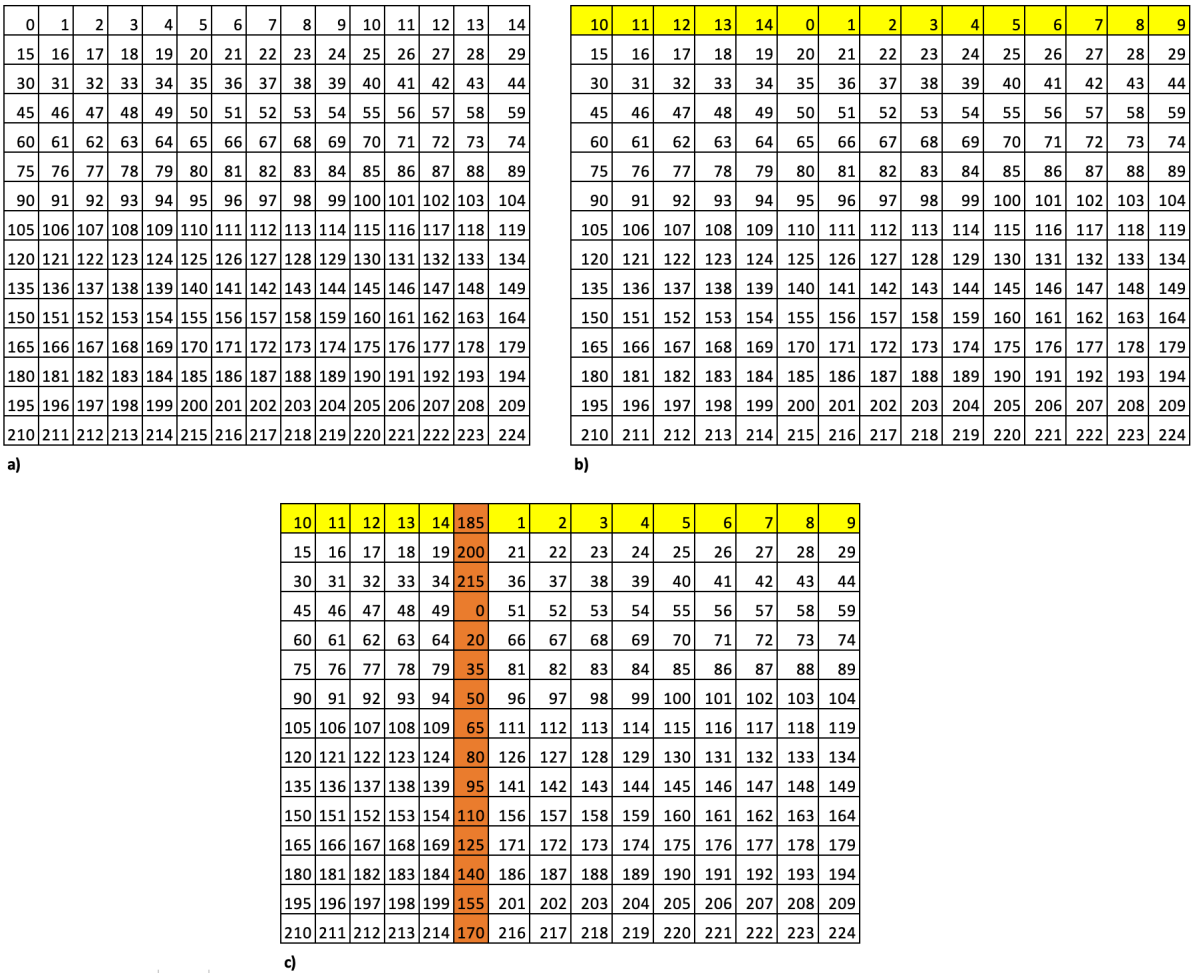
ile birlikte kullanılarak ikinci lojistik haritanın başlangıç parametreleri belirlenir. Birinci lojistik haritanın başlangıç parametreleri ise doğrudan gizli anahtardan türetilir. Şifreleme işlemi *n* satır *m* sütunluk bir matris üzerinde yapılır. İki lojistik harita için de başlangıç parametreleri belirlendikten sonra birinci lojistik haritadan satır bazlı rotasyon için satır sayısı *(n)* kadar rastgele değer üretilir. Bu üretilen değerlerin *[0 – m)* aralığında olması gerekmektedir. Bu nedenle üretilen nümerik değerlerin kolon sayısına *(m)* göre modu alınır. Bu durum Formül (3.1)'de gösterilmiştir.

$$rotationCountForRowIndex = generatedValue \,\%\, m \tag{3.1}$$

Burada *m* sütun sayısını, *generatedValue* ilgili satır için birinci lojistik harita tarafından üretilen rastgele nümerik değeri ve *rotationCountForRowIndex* ilgili satır için rotasyon işleminin kaç adım uygulanacağını gösterir. Resim dosyasındaki ilgili satıra ait piksel değerleri *rotationCountForRowIndex* değeri kadar soldan sağa doğru kaydırılır. Yukarıda bahsedilen adımların benzeri sütun kaydırma işlemleri için de uygulanır. Birinci kaotik haritadan sütun sayısı *(m)* kadar rastgele değer üretilir. Bu üretilen değerlerin *[0 – n)* aralığında olması için üretilen her bir değerin satır sayısına (n) göre modu alınır. Bu durum Formül (3.2)'de gösterilmiştir.

$$rotationCountForColumnIndex = generatedValue \,\%\, n \tag{3.2}$$

Burada *n* satır sayısını, *generatedValue* ilgili sütun için birinci lojistik harita tarafından üretilen rastgele nümerik değeri ve *rotationCountForColumnIndex* ilgili sütun için rotasyon işleminin kaç adım uygulanacağını gösterir. Resim dosyasındaki ilgili sütuna ait piksel değerleri *rotationCountForColumnIndex* değeri kadar yukardan aşağıya doğru kaydırılır. Şekil 3.2'de satır ve sütun bazlı rotasyon işlemi *15x15*'lik bir matris üzerinden adım adım gösterilmiştir. Birinci aşamada ilk satıra 5 adım soldan sağa rotasyon uygulanmıştır. İkinci aşamada birinci aşamada elde edilen yeni matrisin altıncı kolonuna yukarıdan aşağıya doğru 3 adım rotasyon işlemi uygulanmıştır.



a)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69	70	71	72	73	74
75	76	77	78	79	80	81	82	83	84	85	86	87	88	89
90	91	92	93	94	95	96	97	98	99	100	101	102	103	104
105	106	107	108	109	110	111	112	113	114	115	116	117	118	119
120	121	122	123	124	125	126	127	128	129	130	131	132	133	134
135	136	137	138	139	140	141	142	143	144	145	146	147	148	149
150	151	152	153	154	155	156	157	158	159	160	161	162	163	164
165	166	167	168	169	170	171	172	173	174	175	176	177	178	179
180	181	182	183	184	185	186	187	188	189	190	191	192	193	194
195	196	197	198	199	200	201	202	203	204	205	206	207	208	209
210	211	212	213	214	215	216	217	218	219	220	221	222	223	224

b)

10	11	12	13	14	0	1	2	3	4	5	6	7	8	9
15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69	70	71	72	73	74
75	76	77	78	79	80	81	82	83	84	85	86	87	88	89
90	91	92	93	94	95	96	97	98	99	100	101	102	103	104
105	106	107	108	109	110	111	112	113	114	115	116	117	118	119
120	121	122	123	124	125	126	127	128	129	130	131	132	133	134
135	136	137	138	139	140	141	142	143	144	145	146	147	148	149
150	151	152	153	154	155	156	157	158	159	160	161	162	163	164
165	166	167	168	169	170	171	172	173	174	175	176	177	178	179
180	181	182	183	184	185	186	187	188	189	190	191	192	193	194
195	196	197	198	199	200	201	202	203	204	205	206	207	208	209
210	211	212	213	214	215	216	217	218	219	220	221	222	223	224

c)

10	11	12	13	14	185	1	2	3	4	5	6	7	8	9
15	16	17	18	19	200	21	22	23	24	25	26	27	28	29
30	31	32	33	34	215	36	37	38	39	40	41	42	43	44
45	46	47	48	49	0	51	52	53	54	55	56	57	58	59
60	61	62	63	64	20	66	67	68	69	70	71	72	73	74
75	76	77	78	79	35	81	82	83	84	85	86	87	88	89
90	91	92	93	94	50	96	97	98	99	100	101	102	103	104
105	106	107	108	109	65	111	112	113	114	115	116	117	118	119
120	121	122	123	124	80	126	127	128	129	130	131	132	133	134
135	136	137	138	139	95	141	142	143	144	145	146	147	148	149
150	151	152	153	154	110	156	157	158	159	160	161	162	163	164
165	166	167	168	169	125	171	172	173	174	175	176	177	178	179
180	181	182	183	184	140	186	187	188	189	190	191	192	193	194
195	196	197	198	199	155	201	202	203	204	205	206	207	208	209
210	211	212	213	214	170	216	217	218	219	220	221	222	223	224

Şekil 3.2: Satır ve sütun bazlı döngüsel rotasyon örneği.

Bu şekilde karıştırma işlemi matrisin bütün satır ve sütunlarına uygulanır. Karıştırma işlemi tamamlandıktan sonra ikinci lojistik haritadan (gizli anahtar ve özet kod ile başlangıç parametreleri belirlenen kaotik harita) *n x m* boyutlarında rastgele değerlerden oluşan yeni bir matris elde edilir. Daha sonra karıştırılmış matris ile üretilen ikinci matrise ait değerler XoR işlemine tabi tutularak asıl şifreleme işlemi bu aşamada yapılmış olur.

Son aşamada ise ikinci lojistik haritanın başlangıç parametrelerinin belirlenmesinde kullanılan ve orijinal dosyaya ait özet kodun ilk iki byte'ını içeren verinin şifrelenmiş resim dosyası içerisine gömülmesi adıma geçilir. İlk aşamada, orijinal dosyanın *(P)* SHA-224 özet fonksiyonu kullanılarak özet değeri *(H)* elde edilir. Bu özet değerin ilk iki byte'ı alınır ve *H2* olarak isimlendirilir. *H2* değerine bir byte'lık doğrulama değeri eklenerek 3 byte'lık *HV3* değeri elde edilir. Doğrulama değeri özellikle kesme saldırıları ve gürültü saldırılarında özet kodun sağlık bir şekilde elde edilebilmesi için elzemdir ve şifrelenmiş matris içerisinde asıl aklanacak

olan bu *HV3* değeridir. Orijinal dosyadan *HV3* değerinin üretilmesine ait bütün adımlar Formül (3.3), Formül (3.4) ve Formül (3.5)'de gösterilmiştir.

$$H = hashFunctionSHA224(P) \tag{3.3}$$

$$H2 = subString(H, 0, 2) \tag{3.4}$$

$$HV3 = concat(H2, (int(H2[0]) + int(H2[1]) + 1)) \tag{3.5}$$

Bu şekilde hesaplanan *HV3* veri bloğunu şifrelenmiş resim dosyasına dağıtmak için öncelikle resim dosyasının sonuna yeni bir satır eklenir. Bu boş satırın tamamını dolduracak şekilde *HV3* değeri tekrarlanarak yazılır. Matrisin sütun sayısı *90* olduğunu düşünelim, bu durumda *30* kez *(90 / 3 = 30) HV3* değeri art arda yeni eklenen satıra yazılacaktır. Daha sonra birinci lojistik haritadan elde edilen *m* adet veri ile yeni eklenen satır XoR işlemine tabi tutulur. Bu şekilde *HV3* değerlerinin güvenliği sağlanmış olur. Son aşamada ise yine birinci kaotik haritadan *m/3* adet ikili değerler elde edilir. Bu ikili değerler değiş-tokuş (swap) işleminde kullanılacak matris içerisindeki indisleri ifade etmektedir ve bu değerlere göre son satırda bulunan şifrelenmiş veri yine şifrelenmiş matris içerisine dağıtılır. Bu son işlemin çıktısı ile nihai şifrelenmiş dosyayı elde etmiş oluruz.

Şekil 3.3'de örnek bir matris üzerinde şifreleme adımları ayrıntılı bir şekilde sunulmuştur. İlk aşamada *15x15*'lik bir matris oluşturulmuş ve *[0-244]* aralığında değerler başlangıç değeri olarak girilmiştir. Birinci matris orijinal veriyi içermektedir. İkinci matriste birinci kaotik haritadan elde edilen veriler ile satırlar soldan sağa doğru rotasyon işlemine tabi tutulmuştur. Üçüncü matriste ise yine birinci kaotik haritadan elde edilen veriler ile yukarıdan aşağıya doğru sütunlar rotasyon işlemine tabi tutulmuştur. Bu şekilde karıştırma işlemi tamamlanmıştır.

Orijinal matrisin özet değeri SHA2-224 özet fonksiyonu kullanılarak elde edilmiştir. Elde edilen özet değerin ilk iki byte'ı ile gizli anahtar değerleri birlikte kullanılarak ikinci kaotik haritanın başlangıç parametreleri belirlenmiştir. Bu şekilde başlatılan ikinci kaotik haritadan

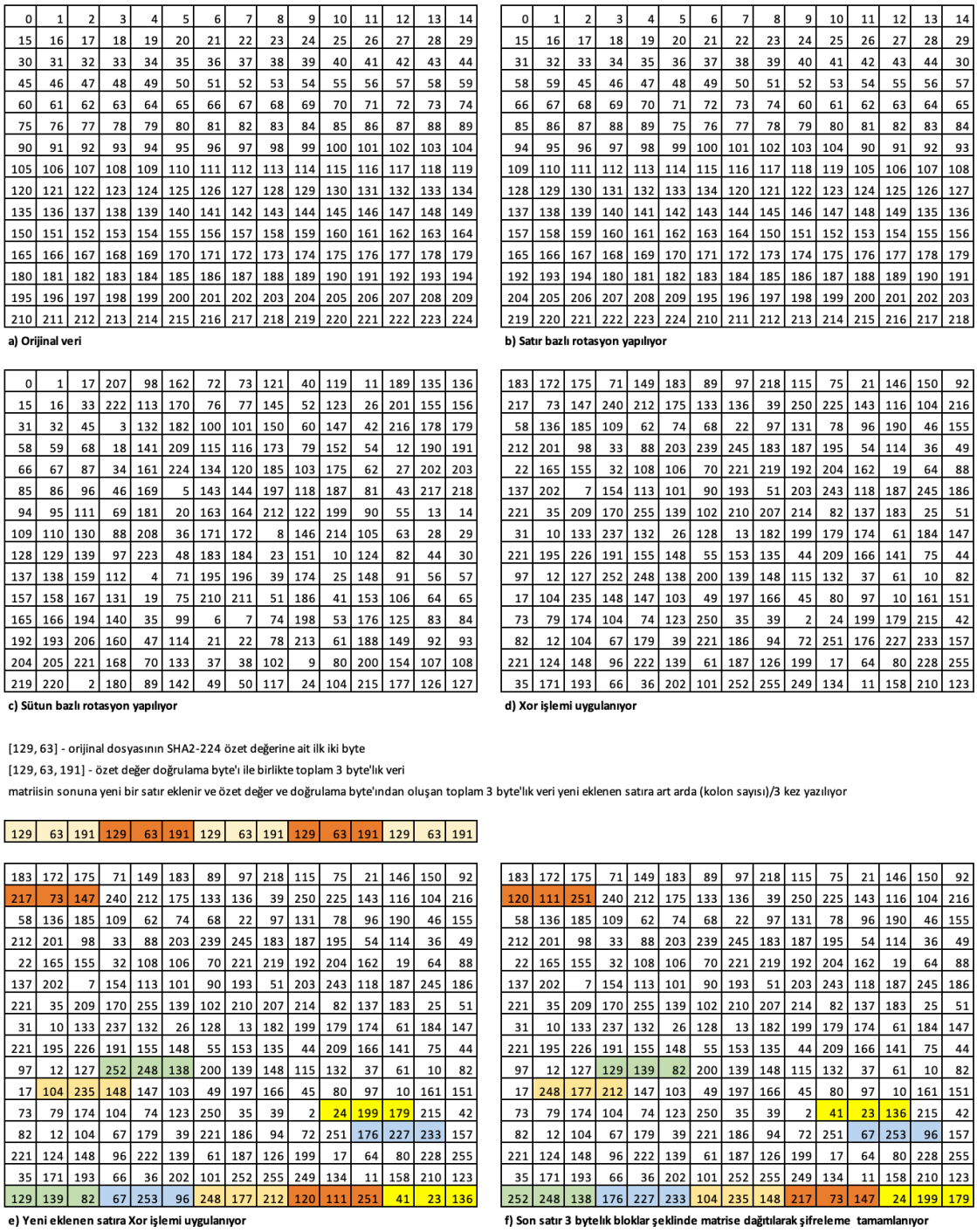
elde edilen rastgele değer ile üçüncü matris XoR işlemine tabi tutulmuş ve dördüncü şifrelenmiş matris elde edilmiştir.

Beşinci adımda matrisin sonuna yeni bir satır eklenmiştir. Gürültü ve kesme saldırılarına karşı koyabilmek için iki byte'lık özet koda bir byte'lık bir doğrulama değeri eklenmiş ve şekilde üç byte'lık bir veri bloğu oluşturulmuştur. Bu üç byte'lık veri bloğu yeni eklenen satıra tekrarlanan bir biçimde art arda yazılmıştır. Daha sonra bu son satır değerleri birinci kaotik haritadan elde edilen değerler ile XoR'lanmıştır. Bu duruma ait çıktı beşinci matriste gösterilmiştir.

Altıncı matriste ise, birinci kaotik haritadan elde edilen rastgele değerler ile (bu değerler orijinal dosyanın satır ve sütun boyutlarına indirgenmiştir) son satır verisi üç byte'lık bloklar şeklinde matris içerisine dağıtılmış (değiş tokuş işlemi uygulanmış) ve bu sayede nihai şifrelenmiş veri elde edilmiştir.

Şekil 3.4'de geliştirilen şifreleme algoritması kullanılarak *512x512* gri tonlamalı "Lena.png" resim dosyasının şifrelenmesi adımları aşama aşama gösterilmiştir. Birinci resim dosyası orijinal veriyi göstermektedir. İkinci dosya orijinal resim dosyasının satır bazlı rotasyon işleminde sonra elde edilmiştir. Üçüncü dosya, ikinci adımda elde edilen satır bazlı rotasyon işlemini sonucu elde edilen dosyaya sütun bazlı rotasyon işlemi uygulanarak elde edilmiştir. Dördüncü adımda verilen dosya üçüncü adımın çıktısına XoR işlemi uygulanarak elde edilmiştir. Son aşamada ise, dördüncü adımda elde edilen çıktıya yeni bir satırın eklenmesi, özet değerin ilk iki byte'ı ve doğrulama byte olmak üzere toplam 3 byte'lık verinin bu satıra tekrarlanarak yazılması ve sonrasında şifrelenerek resim dosyası üzerine dağıtılması ile elde edilmiş nihai şifrelenmiş resim dosyası verilmiştir.

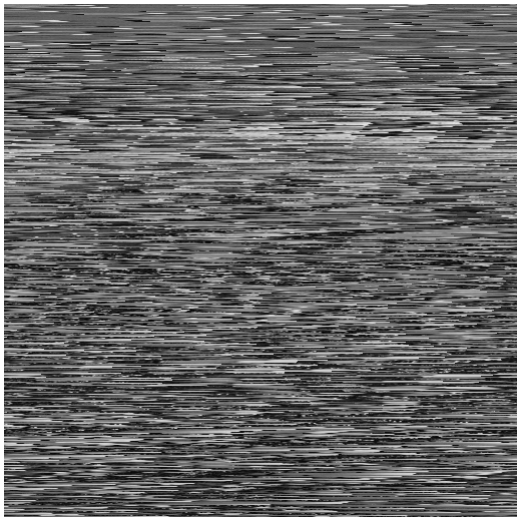
Şekil 3.5'de şifreleme algoritması akış diyagramı detaylı bir şekilde verilmiştir. Kullanılan kaotik haritalar, bu haritaların hangi parametreler ile nasıl başlatıldıkları, bu haritalardan elde edilen çıktıların nasıl ve hangi aşamalarda kullanıldıkları, rotasyon işlemleri girdi ve çıktıları, XoR işlemleri, adım adım gösterilmiştir. Ayrıca her bir adıma ait girdi ve çıktı diyagramda verilmiştir. Bu akış diyagramına karşılık gelen kaba kod ise Algoritma 1 – Algoritma 3 aralığında verilmiştir.



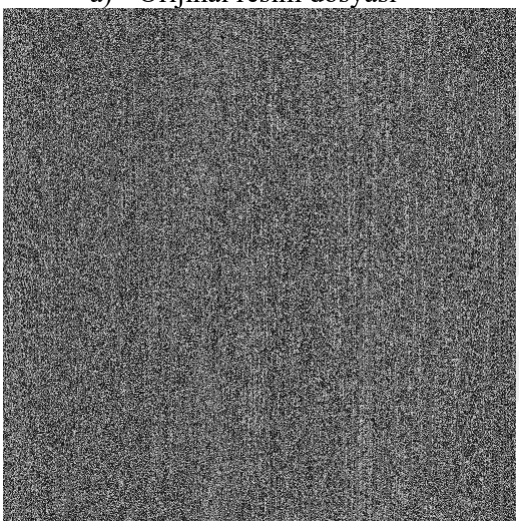
Şekil 3.3: Örnek bir matris üzerinde şifreleme adımları.

a) Orijinal dosya, b) Satır bazlı rotasyon işlemi uygulandıktan sonra elde edilen çıktı, c) Sütun bazlı rotasyon işlemi uygulandıktan sonra elde edilen çıktı, d) XoR işlemi uygulandıktan sonra elde edilen çıktı, e) Yeni satır eklendikten ve XoR'lu tekrarlanan özet kod eklendikten sonra elde edilen çıktı, f) Son satır matris üzerine dağıtıldıktan sonra elde edilen çıktıyı göstermektedir.

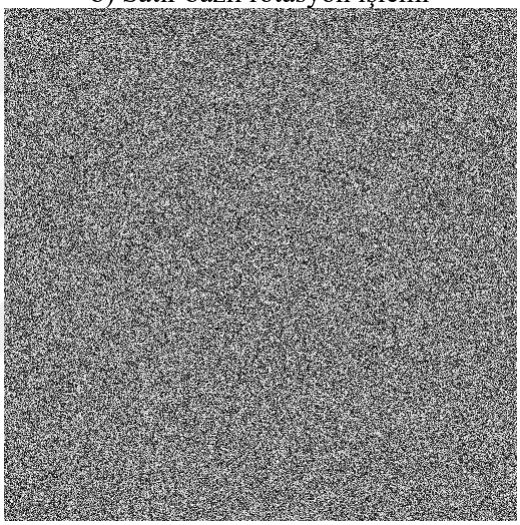
a) Orijinal resim dosyası



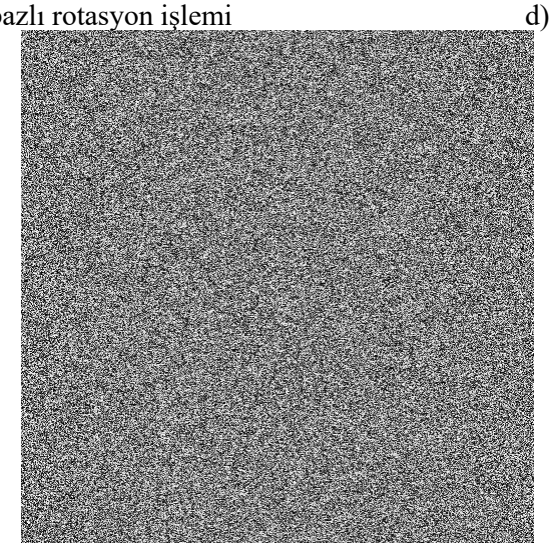
b) Satır bazlı rotasyon işlemi



c) Sütun bazlı rotasyon işlemi

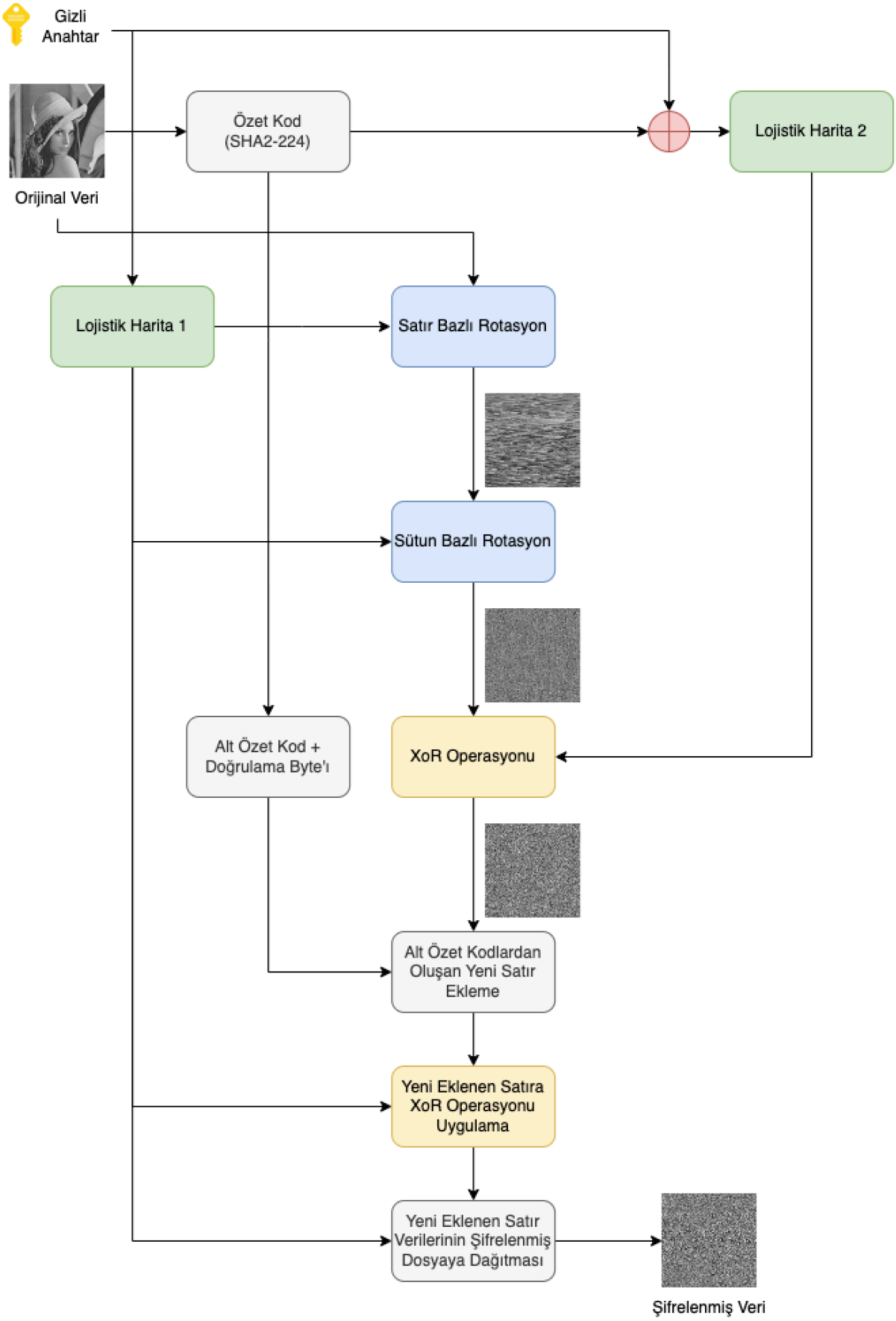


d) XoR işlemi



e) Şifrelenmiş resim dosyası

Şekil 3.4: Şifreleme adımlarının resim dosyası üzerinden analizi.



Şekil 3.5: Şifreleme akış diyagramı.

Algoritma 1 Şifreleme kaba kodu

```
1:  procedure encryption (S, P)
2:      n, m ← getSize(P)                          //get matrix size
3:      H ← calculateHashValue(P)                  //get hash value of plaintext
4:      H2 ← getFirstTwoBytes(H)                   //get only fist two bytes of hash value
5:      HV3 ← getValidatedH2(H2)                   //2 bytes H2 value + 1 byte validation
6:      C^1 ← confusion(S, P)                      //confusion phase
7:      C^2 ← diffusion(S, C^1, H2)                //diffusion phase
8:      C^3 ← addNewRowToBottom(C^2)               //add new row at the bottom of the matrix
9:      for i ← 1 to m/3 do                        //since HV3 is 3 bytes
10:         C^3[n, 3i] ← HV3[0]                    //fill bottom line with repetitive HV3 value
11:         C^3[n, 3i + 1] ← HV3[1]                as column size / 3 times
12:         C^3[n, 3i + 2] ← HV3[2]
13:     end for
14:     K4X ← getKeySequenceFromLogicMap1(S, m)    //key stream for bottom row XoR
15:     for i ← 1 to m do
16:         C^3[n, i] ← XoR(C^3[n, i], K4X[i])     //encrypt bottom row
17:     end for
18:     K4S ← getKeySequenceFromLogicMap1(S, 2m/3) //key stream for swap operation
19:     for i ← 1 to m/3 do
20:         C^3[n, 3i] ↔ C^3[K4S[2i], K4S[2i+1]]   //bottom row data distributed into matrix
21.         C^3[n, 3i+1] ↔ C^3[K4S[2i], K4S[2i+1]+1] by swapping operation
22.         C^3[n, 3i+2] ↔ C^3[K4S[2i], K4S[2i+1]+2]
23:     end for
24:     C ← C^3                                    //final ciphertext
25: end procedure
```

Algoritma 2 Karıştırma kaba kodu

```
1:  procedure confusion (S, P)
2:      n, m ← getSize(P)                          //get matrix size
3:      K4RR ← getKeySequenceFromLogicMap1(S, m)   //key stream for row rotation
4:      for i ← 1 to n do
5:          C^I ← rowBasedCyclicRotation(P, K4RR[i])
6:      end for
7:      K4CR ← getKeySequenceFromLogicMap1(S, n)   //key stream for column rotation
8:      for i ← 1 to m do
9:          C^II ← columnBasedCyclicRotation(C^I, K4CR[i])
10:     end for
11: end procedure
```

Algoritma 3 Dağıtma kaba kodu

```
1:  procedure diffusion (S, C^1, H2)
2:      n, m ← getSize (C^1)                       //get matrix size
3:      K4X ← getKeySequenceFromLogicMap2(S, H2, n*m) //key stream for XoR operation
4:      for i ← 1 to n do
5:          for j ← 1 to m do
6:              C^1 ← XoR(C^1, K4X[i, j])          //apply XoR operation
7:          end for
8:      end for
9:  end procedure
```

Şekil 3.6: Önerilen yöntemin şifreleme algoritmaları.

3.2.2. Şifre Çözme Adımları

Şifre çözme (decryption) aşamasında şifreleme aşamasında uygulanan adımlar ters sırada tek tek uygulanmıştır. Öncelikle şifreli resim dosyasından 2 byte'lık *H2* değerinin elde edilmesi gerekmektedir. Bunun için yalnızca gizli anahtar kullanılarak başlatılan birinci lojistik haritadan elde edilen rastgele veriler ile şifrelenmiş resim dosyası içerisine şifreli ve rastgele bir şekilde dağıtılmış olan 3 byte'lık veri blokları (*HV3*) resim dosyasının en alt satırında toplanmıştır. Daha sonra yine birinci lojistik haritadan elde edilen rastgele veriler ile resmin son satırına XoR işlemi uygulanarak şifreli değerler çözülmüştür.

Son satırda toplanan ve tekrarlanan *HV3* veri bloğundan *H2* değerinin elde edilmesi gerekmektedir. Bunun için öncelikli olarak son satır verileri 3 byte'lık bloklar şeklinde parçalara ayrılmıştır. Her bir parça için Formül (3.5)'de verilen doğrula işlemi uygulanmıştır. Doğrulama işlemine uyan bu *HV3* değerlerine ait son byte iptal edilerek 2 byte'lık bloklar şeklinde alınıp kullanım frekanslarına bakılmıştır. Kullanım frekansı en yüksek olan 2 byte'lık veri bloğu *H2* değeri olarak belirlenmiştir. Bir sonraki aşamada ise buradan elde edilen bu 2 byte'lık *H2* değeri gizli anahtar ile birlikte kullanılarak ikinci kaotik haritanın başlangıç parametrelerinin belirlenmiştir. *H2* değeri belirlendikten sonra resim dosyasına ait son satır silinmiş ve orijinal şifrelenmiş dosya elde edilmiştir. İkinci lojistik haritadan *n x m* boyutlarındaki rastgele değerlerden oluşan bir matris üretilmiş ve bu matris ile şifrelenmiş veri XoR işlemine tabi tutulmuştur.

Son aşamada XoR işlemi ile çözülen veri rotasyon işlemleri ile düzenlenmiştir. İlk olarak birinci lojistik haritadan sütun sayısı kadar *(m)* ve *[0 - n)* aralığında değerler üretilmiş ve her bir sütun, ilgili sütun için üretilen rotasyon değeri kadar aşağıdan yukarıya doğru rotasyon işlemine tabi tutulmuştur. Son adım olarak buradan elde edilen sütunları düzenlenmiş veri için satır bazlı düzenleme işlemi yapılmıştır. Bu işlem için satır sayısı kadar *(n)* ve *[0 - m)* aralığında rastgele değerler birinci kaotik haritadan elde edilmiştir. Her bir satır kendisi için üretilen rastgele değer kadar sağdan sola rotasyon işlemine tabi tutulmuştur. Bu son işlem ile birlikte artık orijinal veriye ulaşılmış ve şifre çözeme işlemi tamamlanmıştır.

Şekil 3.3'de örnek bir matris üzerinde gösterilen şifreleme adımları Şekil 3.7'da ters yönde şifre çözeme olarak uygulamıştır. Her bir aşama ayrıntılı bir şekilde adım adım gösterilmiştir. Her iki örnekte de aynı veri seti kullanılmıştır. İlk matriste (yani girdi olarak) şifreleme adımında elde edilen nihai çıktı verilmiştir. İkinci matriste, matris içerisine dağıtılmış

HV3 veri bloklarının matrisin son satırında toplanması gösterilmektedir. Üçüncü matriste ise son satırda toplanan *HV3* veri bloklarından *H2* değeri elde edildikten sonra kalan orijinal şifrelenmiş veri gösterilmiştir. Dördüncü adımda ise gizli anahtar ve *H2* değeri kullanılarak başlatılan ikinci kaotik haritadan elde edilen *n x m* rastgele değerlerden oluşan matris ile şifrelenmiş matris XoR işlemine tabi tutulmuş ve elde edilen çıktı verilmiştir. Beşinci aşamada sütun bazlı tersine rotasyon işlemi uygulandıktan sonra elde edilen çıktı verilmiştir. Son aşamada satır bazlı tersine rotasyon sonrası elde edilen nihai matris sunulmuştur. Son aşamada elde edilen bu matris şifreleme aşamasında kullanılan orijinal matrisin aynısıdır. Bu da algoritmanın şifreleme ve şifre çözme adımlarını başarılı bir şekilde gerçekleştirdiğini göstermektedir.

Şekil 3.8'de şifre çözme algoritmasının gerçek resim dosyası üzerindeki adım adım aşamaları sunulmuştur. Şifreleme adımında kullanılan *512x512* "Lena.png" gri tonlamalı resim dosyasına ait şifrelenmiş veri bu aşamada girdi olarak kullanılmıştır. Şifre çözme adımlarının sonunda orijinal veri *512x512* "Lena.png" gri tonlamalı resim dosyası başarılı bir şekilde elde edilmiştir. Şifreleme aşamasında kullanılan orijinal dosya ile şifre çözme aşamasında elde dilen dosya En Yüksek Sinyal-Gürültü Oranı (Peak Signal-to-Noise Ratio - PSNR) kullanılarak karşılaştırılmış. Aynı iki dosya için PSNR değeri sonsuz olarak hesaplanır ve bu karşılaştırmada PSNR değeri sonsuz olarak elde edilmiştir. Yani şifre çözme adımı sonunda herhangi bir kayıp olmadan orijinal dosyanın elde edildiğinden emin olunmuştur.

Şekil 3.9'de şifre çözme algoritması akış diyagramı detaylı bir şekilde verilmiştir. Kullanılan kaotik haritalar, bu haritaların hangi parametreler ile nasıl başlatıldıkları, bu haritalardan elde edilen çıktıların hangi aşamalarda nasıl kullanıldıkları, rotasyon işlemleri girdileri ve çıktıları, XoR işlemleri, aşama aşama gösterilmiştir. Ayrıca her bir aşamanın girdisi ve çıktısı diyagramda verilmiştir. Şifreleme diyagramı ile şifre çözme diyagramları detaylı bir şekilde incelendiğinde görüleceği üzere şifreleme aşamasında uygulanan her bir adım şifre çözme aşamasında tersten uygulanmış ve bu sayede şifre çözme işlemi başarılı bir şekilde tamamlanmıştır.

183	172	175	71	149	183	89	97	218	115	75	21	146	150	92
120	111	251	240	212	175	133	136	39	250	225	143	116	104	216
58	136	185	109	62	74	68	22	97	131	78	96	190	46	155
212	201	98	33	88	203	239	245	183	187	195	54	114	36	49
22	165	155	32	108	106	70	221	219	192	204	162	19	64	88
137	202	7	154	113	101	90	193	51	203	243	118	187	245	186
221	35	209	170	255	139	102	210	207	214	82	137	183	25	51
31	10	133	237	132	26	128	13	182	199	179	174	61	184	147
221	195	226	191	155	148	55	153	135	44	209	166	141	75	44
97	12	127	129	139	82	200	139	148	115	132	37	61	10	82
17	248	177	212	147	103	49	197	166	45	80	97	10	161	151
73	79	174	104	74	123	250	35	39	2	41	23	136	215	42
82	12	104	67	179	39	221	186	94	72	251	67	253	96	157
221	124	148	96	222	139	61	187	126	199	17	64	80	228	255
35	171	193	66	36	202	101	252	255	249	134	11	158	210	123
252	248	138	176	227	233	104	235	148	217	73	147	24	199	179

a) Şifreli veri

183	172	175	71	149	183	89	97	218	115	75	21	146	150	92
217	73	147	240	212	175	133	136	39	250	225	143	116	104	216
58	136	185	109	62	74	68	22	97	131	78	96	190	46	155
212	201	98	33	88	203	239	245	183	187	195	54	114	36	49
22	165	155	32	108	106	70	221	219	192	204	162	19	64	88
137	202	7	154	113	101	90	193	51	203	243	118	187	245	186
221	35	209	170	255	139	102	210	207	214	82	137	183	25	51
31	10	133	237	132	26	128	13	182	199	179	174	61	184	147
221	195	226	191	155	148	55	153	135	44	209	166	141	75	44
97	12	127	252	248	138	200	139	148	115	132	37	61	10	82
17	104	235	148	147	103	49	197	166	45	80	97	10	161	151
73	79	174	104	74	123	250	35	39	2	24	199	179	215	42
82	12	104	67	179	39	221	186	94	72	251	176	227	233	157
221	124	148	96	222	139	61	187	126	199	17	64	80	228	255
35	171	193	66	36	202	101	252	255	249	134	11	158	210	123
129	139	82	67	253	96	248	177	212	120	111	251	41	23	136

b) Xor'lu özet değerler en alt satırda toplanıyor

[9,3], [12,11], [10,1], [1,0], [11,10] indekslerinde bulunan 3 byte'lık veriler ile son satırda bulunan veriler değiş tokuş yapılıyor
Son satırda toplanan Xor'lu 3 byte'lık bloklar önce Xor işlemi ile açılıyor

129	63	191	129	63	191	129	63	191	129	63	191	129	63	191

Doğrulama kodu ile kontrol edilen ve başarılı olan 2 byte'lık veri bloklarından en çok kullanılan blok özet değerin ilk iki byte'ı olarak belirleniyor
[129, 63, 191] - kurala uyan 3 byte'lık veri bloğu
[129, 63] - özet değerin ilk byte'ı olarak tespit ediliyor

183	172	175	71	149	183	89	97	218	115	75	21	146	150	92
217	73	147	240	212	175	133	136	39	250	225	143	116	104	216
58	136	185	109	62	74	68	22	97	131	78	96	190	46	155
212	201	98	33	88	203	239	245	183	187	195	54	114	36	49
22	165	155	32	108	106	70	221	219	192	204	162	19	64	88
137	202	7	154	113	101	90	193	51	203	243	118	187	245	186
221	35	209	170	255	139	102	210	207	214	82	137	183	25	51
31	10	133	237	132	26	128	13	182	199	179	174	61	184	147
221	195	226	191	155	148	55	153	135	44	209	166	141	75	44
97	12	127	252	248	138	200	139	148	115	132	37	61	10	82
17	104	235	148	147	103	49	197	166	45	80	97	10	161	151
73	79	174	104	74	123	250	35	39	2	24	199	179	215	42
82	12	104	67	179	39	221	186	94	72	251	176	227	233	157
221	124	148	96	222	139	61	187	126	199	17	64	80	228	255
35	171	193	66	36	202	101	252	255	249	134	11	158	210	123

c) Özet değer elde edildikten sonra son satır siliniyor

0	1	17	207	98	162	72	73	121	40	119	11	189	135	136
15	16	33	222	113	170	76	77	145	52	123	26	201	155	156
31	32	45	3	132	182	100	101	150	60	147	42	216	178	179
58	59	68	18	141	209	115	116	173	79	152	54	12	190	191
66	67	87	34	161	224	134	120	185	103	175	62	27	202	203
85	86	96	46	169	5	143	144	197	118	187	81	43	217	218
94	95	111	69	181	20	163	164	212	122	199	90	55	13	14
109	110	130	88	208	36	171	172	8	146	214	105	63	28	29
128	129	139	97	223	48	183	184	23	151	10	124	82	44	30
137	138	159	112	4	71	195	196	39	174	25	148	91	56	57
157	158	167	131	19	75	210	211	51	186	41	153	106	64	65
165	166	194	140	35	99	6	7	74	198	53	176	125	83	84
192	193	206	160	47	114	21	22	78	213	61	188	149	92	93
204	205	221	168	70	133	37	38	102	9	80	200	154	107	108
219	220	2	180	89	142	49	50	117	24	104	215	177	126	127

d) Xor işlemi uygulanıyor

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
31	32	33	34	35	36	37	38	39	40	41	42	43	44	30
58	59	45	46	47	48	49	50	51	52	53	54	55	56	57
66	67	68	69	70	71	72	73	74	60	61	62	63	64	65
85	86	87	88	89	75	76	77	78	79	80	81	82	83	84
94	95	96	97	98	99	100	101	102	103	104	90	91	92	93
109	110	111	112	113	114	115	116	117	118	119	105	106	107	108
128	129	130	131	132	133	134	120	121	122	123	124	125	126	127
137	138	139	140	141	142	143	144	145	146	147	148	149	135	136
157	158	159	160	161	162	163	164	150	151	152	153	154	155	156
165	166	167	168	169	170	171	172	173	174	175	176	177	178	179
192	193	194	180	181	182	183	184	185	186	187	188	189	190	191
204	205	206	207	208	209	195	196	197	198	199	200	201	202	203
219	220	221	222	223	224	210	211	212	213	214	215	216	217	218

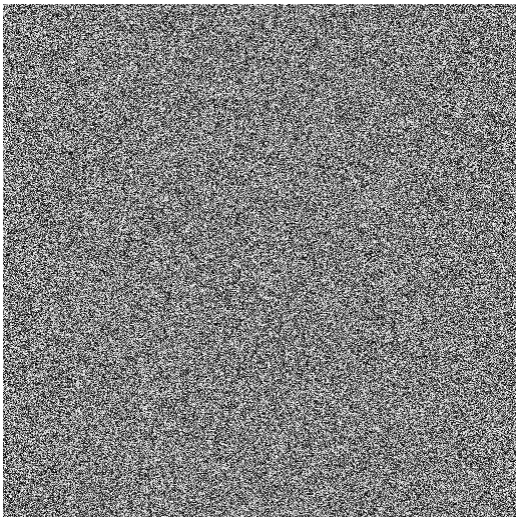
e) Sütun bazlı tersine rotasyon uygulanıyor

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69	70	71	72	73	74
75	76	77	78	79	80	81	82	83	84	85	86	87	88	89
90	91	92	93	94	95	96	97	98	99	100	101	102	103	104
105	106	107	108	109	110	111	112	113	114	115	116	117	118	119
120	121	122	123	124	125	126	127	128	129	130	131	132	133	134
135	136	137	138	139	140	141	142	143	144	145	146	147	148	149
150	151	152	153	154	155	156	157	158	159	160	161	162	163	164
165	166	167	168	169	170	171	172	173	174	175	176	177	178	179
180	181	182	183	184	185	186	187	188	189	190	191	192	193	194
195	196	197	198	199	200	201	202	203	204	205	206	207	208	209
210	211	212	213	214	215	216	217	218	219	220	221	222	223	224

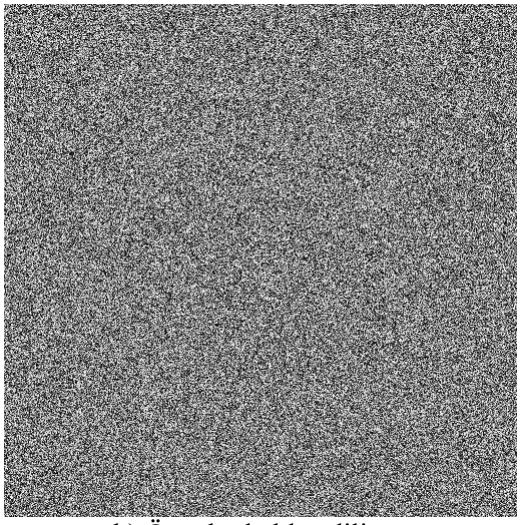
f) Satır bazlı tersine rotasyon işlemi uygulanarak orijinal veri elde ediliyor

Şekil 3.7: Örnek bir matris üzerinde şifre açma adımları.

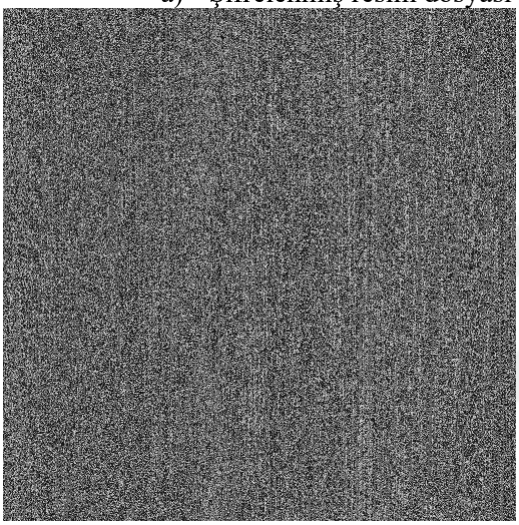
a) Şifreli veri, b) *HV3* değerleri son satırda toplanıyor, c) *H2* elde edildikten sonra son satır siliniyor, d) XoR işlemi uygulanıyor, e) Sütun bazlı ters rotasyon, f) Satır bazlı ters rotasyon.



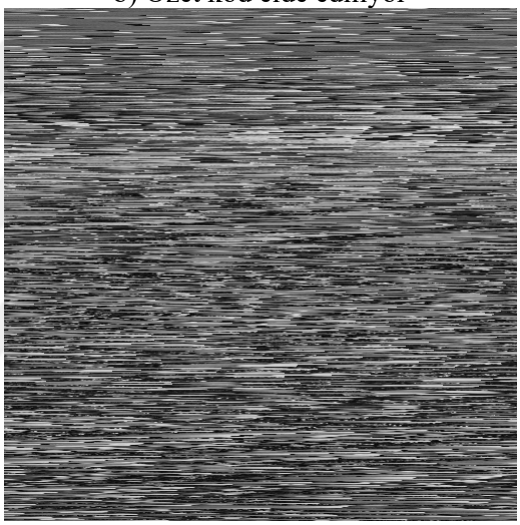
a) Şifrelenmiş resim dosyası



b) Özet kod elde ediliyor



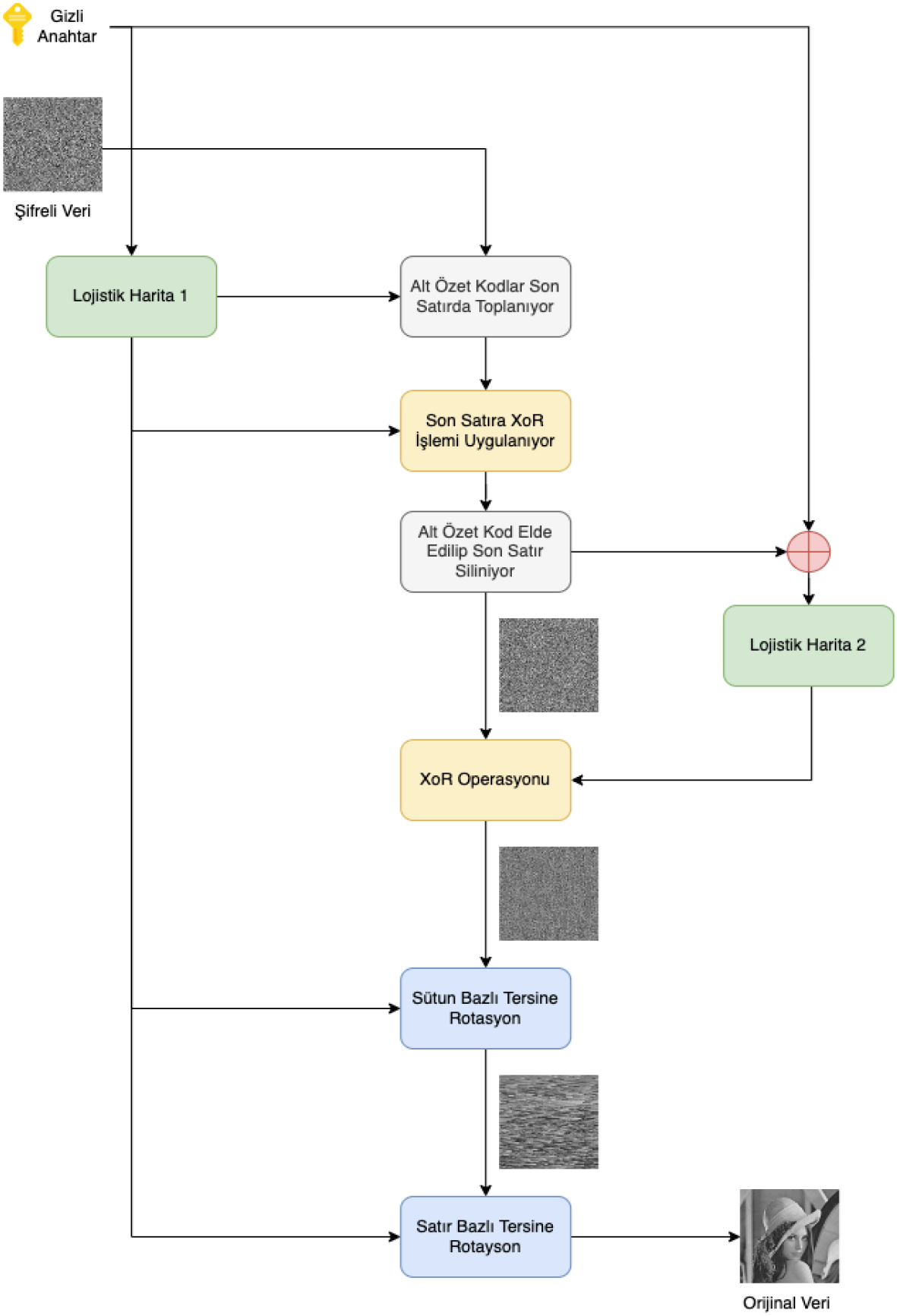
c) XoR işlemi uygulanıyor



d) Tersine sütun bazlı rotasyon

e) Tersine satır bazlı rotasyon sonrası nihai dosya elde ediliyor

Şekil 3.8: Şifre çözme adımlarının resim dosyası üzerinden analizi.



Şekil 3.9: Şifre Çözme Adımları.

3.3. OLASILIKSAL ŞİFRELEME (PROBABILISTIC ENCRYPTION)

Tez kapsamında geliştirmiş olduğumuz şifreleme yapısını temel alarak ikinci bir şifreleme algoritması geliştirilmiştir. Geliştirilen bu yeni algoritmada son zamanların popüler şifreleme yöntemlerinden Olasılıksal Şifreleme (Probabilistic Encryption) yaklaşımı temel alınmış ve mevcut yapıya uyarlanmıştır. Geleneksel şifreleme algoritmaları aynı veriyi aynı anahtar ile şifrelediğimizde aynı şifreli veriyi üretecektir. Bu durum aynı mesajın birden fazla paylaşılması durumunda ciddi bir güvenlik zafiyetinin çıkmasına neden olmaktadır. Olasılıksal Şifreleme (Probabilistic Encryption) bu sorunu çözmek geçiştirilmiş bir şifreleme yöntemidir. Aynı verinin aynı anahtar ile her bir şifrelenme işleminde farklı şifreli verilerin elde edilmesi yöntemine olasılıksal şifreleme denilmektedir. Yukarıda detaylarını paylaşmış olduğumu algoritma yapısına bağlı kalarak yeni bir şifreleme (olasılıksal şifreleme) algoritması tasarlamıştır.

Bu yeni algoritmada rastgele sayı üreteci olarak iki farklı tent haritası kullanılmıştır. İlk tent haritanın başlangıç değerleri doğrudan gizli anahtardan türetilmiş ve üretilen değerler karıştırma işleminde kullanılmıştır. İkinci tent haritanın başlangıç değerleri gizli anahtar ve 2 byte'lık rastgele değerden türetilmiş (bu değer her şifleme işleminde yeninden oluşturulmaktadır ve bu sayede aynı anahtar ve aynı açık veri için farklı farklı şifrelenmiş dosyalar oluşmaktadır) ve üretilen değerler XoR işleminde kullanılmıştır. Karıştırma adımında Knuth – Durstenfeld karıştırma algoritması kullanılmıştır. İkinci haritanın başlangıç değerinde kullanılan rastgele değer, şifre çözme adımında ihtiyaç duyulacağı için şifrelenerek şifreli dosyanın içerisine dağıtılmıştır.

3.3.1. Rastgele Sayı Üreteci

Bu çalışma kapsamında rastgele sayısı üreteci olarak, basit yapısı, kolay uygulanabilirliği, yaygın kullanımı vs. nedenlerden dolayı iki adet bir boyutlu (1-D) tent harita kullanılmıştır. İlk tent haritanın başlangıç değerleri gizli anahtardan türetilmiştir ve buradan üretilen değerler karıştırma (shuffling), rastgele (random) değerin şifrelenmesi (XoR işlemi) ve rastgele değerin resim dosyası içerisine dağıtılmasında kullanılmıştır. İkinci tent haritanın başlangıç değerleri gizli anahtar ve rastgele değer kullanılarak türetilmiştir. Burada üretilen değerler resim dosyasın şifrelenmesinde (XoR işlemi) kullanılmıştır.

3.3.2. Rastgele Değer (Random Value)

2 byte'lık bir değerdir *R2* şeklinde isimlendirilmiştir. Her bir şifreleme işleminde yeninden oluşturmaktadır. Bu sayede aynı veri ve aynı anahtar için her bir şifreleme işleminde farklı farklı şifrelenmiş verilerin oluşması sağlanmıştır. *R2* değeri ikinci kaotik haritanın başlangıç parametrelerinin belirlenmesinde kullanıldığı için şifrelenin verinin her bir şifreleme işleminde tamamen farklılaşmasına neden olmaktadır. *R2* değerine şifre çözme aşamasında tekrar ihtiyaç duyulacağı için şifreli dosya içerisinde bozulmadan güvenli bir şekilde saklanması gerekmektedir. Bu amaçla *R2* değerine 1 byte'lık doğrulama değeri eklenerek 3 byte'lık *RV3* değeri oluşturulup şifreli dosya içerinde saklanmıştır.

3.3.3. Karıştırma Fonksiyonu

Önerilen olasılıksal şifreleme algoritmada karıştırma işlemi için Knuth – Durstenfeld karıştırma algoritması temel alınarak matris uyarlaması geliştirilmiştir. Birinci kaotik haritandan elde edilen anahtar dizi kullanılarak karıştırma işlemi yapılmıştır. Şekil 3.10'da Knuth – Durstenfeld karıştırma algoritmasının "SHUFFLE" sözcüğü üzerinde uyarlanması verilmiştir.

Rastgele Değer	Aralık								
2	1-6		S	H	U	F	F	L	**E**
3	1-5		S	E	U	F	F	**L**	H
1	1-4		S	E	L	F	**F**	U	H
3	1-3		F	E	L	**F**	S	U	H
1	1-2		F	E	**F**	L	S	U	H
			F	E	F	L	S	U	H

Şekil 3.10: Knuth–Durstenfeld karıştırma algoritması uyarlanması.

Bu şekilde komşu pikseller arasındaki bağlantılar kopartılmış, orijinal dosya ile şifrelenmiş dosya pikselleri birbirlerinden konum olarak ayrıştırılmış olmaktadır. Bu bize korelasyon analizinde yatay, dikey ve çapraz korelasyon katsayılarının çok düşük çıkması ve bu şekilde yapılacak saldırıların önlenmesini sağlamaktadır. Ayrıca bu şekilde karıştırma işlemi kırpma saldırılarına karşı algoritmanın karşı koyabilmesi için elzemdir. Kırpma saldırılarında saldırganlar şifrelenmiş dosyanın farklı bölgelerinin tamamen siyah ya da tamamen beyaz olacak şekilde değiştirmekte ve bu şekilde farklılaştırılan dosya şifre çözücü ile açılmaktadır. Açılan dosya üzerinden orijinal veri, gizli anahtar ve şifreleme algoritmasının yapısı hakkında bilgiler sızdırılmaya çalışılmaktadır.

3.3.4. Şifreleme Fonksiyonu

Şifrele fonksiyonu olarak mantıksal işlemlerden XoR işlemi uygulanmıştır. Her bir bit XoR işlemine tabii tutularak şifreleme gerçekleştirilmiştir. İkinci kaotik haritadan üretilen anahtar dizisi ile karıştırılmış veri XoR işlemine tabi tutularak şifreleme gerçekleştirilmiştir. İkinci kaotik haritanın başlangıç değerlerinin belirlenmesinde rastgele değer kullanıldığı için her bir şifreleme adımında farklı bir anahtar dizisi oluşacaktır ve bu durum aynı metin ve aynı anahtar için her bir şifreleme işleminde farklı bir şifreli verinin oluşmasına neden olacaktır. Şifre çözme işleminde de aynı anahtar dizisi ikinci kez XoR işlemine sokulacak ve bu sayede orijinal veri elde edilecektir.

3.4. ÖNERİLEN OLASILIKSAL ŞİFRELEME ALGORİTMASI

Önerilmen olasılıksal şifreleme algoritmasında iki farklı tent harita rastgele sayı üreteci olarak kullanılmıştır. Şifreleme aşamasında rastgele üretilen 2 byte'lık değer ikinci haritanın başlangıç değerlerinin belirlenmesinde bir parametre olarak kullanılmış ve bu sayede diferansiyel saldırılara karşı dirençli bir yapı oluşturulmuştur. Ayrıca aynı metin ve aynı anahtar için her bir şifreleme işleminde farklı bir şifreli metinin oluşması sağlanmıştır. Karıştırma işlemleri için Knuth – Durstenfeld karıştırma algoritmasının matris versiyonu uyarlanmış ve yer değiştirme değerler birinci kaotik haritadan elde edilmiştir. Bütün şifreleme ve şifre çözme adımları aşağıda detaylı bir şekilde açıklanmıştır.

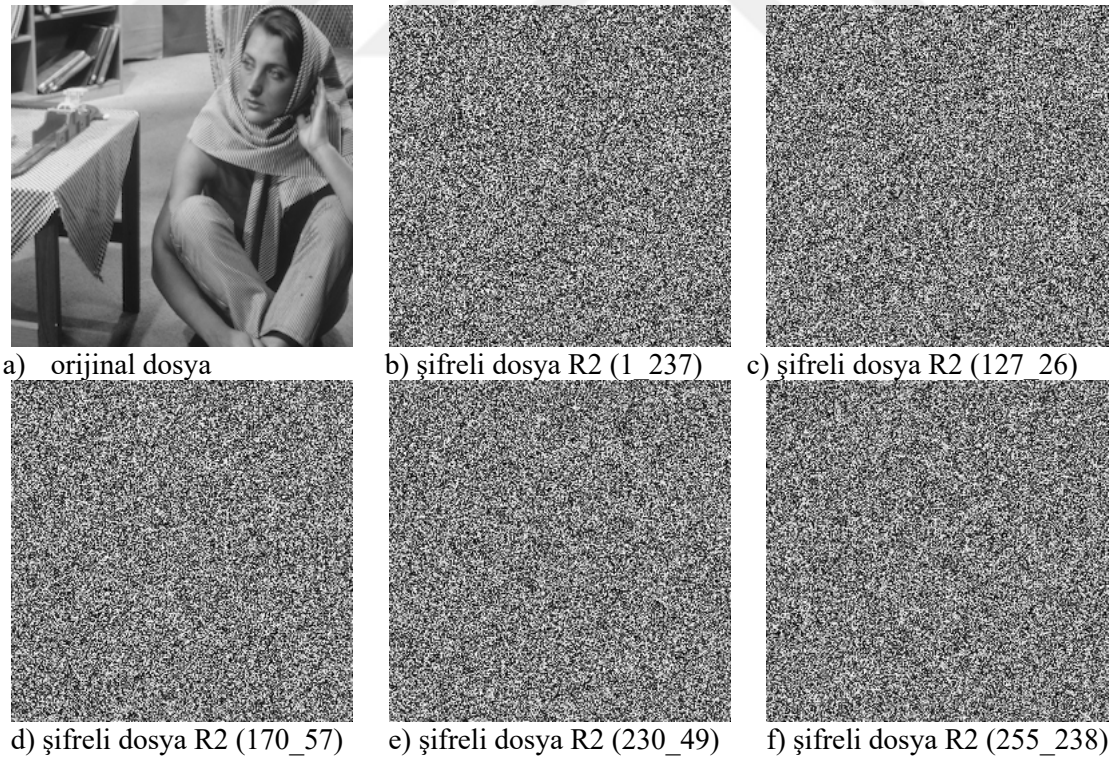
3.4.1. Şifreleme Adımları

Şifreleme işlemine başlamadan önce kullanılan sistem üzerinde 2 byte'lık rastgele bir değer üretilir ve bu değer *R2* olarak isimlendirilir. *R2* değeri gizli anahtar ile birlikte kullanılarak ikinci tent haritanın başlangıç parametreleri belirlenir. Birinci tent haritanın başlangıç parametreleri ise doğrudan gizli anahtardan türetilir. Şifreleme işlemi *n* satır *m* sütunluk bir matris üzerinde yapılır. İki tent harita için de başlangıç parametreleri belirlendikten sonra birinci tent haritadan elde edilen rastgele değerler Knuth – Durstenfeld karıştırma algoritmasının 2 boyutlu uyarlanmasında kullanılarak orijinal veri karıştırılır.

Karıştırma işlemi tamamlandıktan sonra ikinci tent haritadan (gizli anahtar ve *R2* değerleri ile başlangıç parametreleri belirlenen kaotik harita) *n x m* boyutlarında rastgele değerlerden oluşan yeni bir matris elde edilir. Daha sonra karıştırılmış matris ile üretilen ikinci matrise ait değerler XoR işlemine tabi tutularak asıl şifreleme işlemi bu aşamada yapılmış olur.

Son aşamada ise ikinci tent haritanın başlangıç parametrelerinin belirlenmesinde kullanılan *R2* değerin şifrelenmiş resim dosyası içerisine gömülmesi adıma geçilir. *R2* değerine bir byte'lık doğrulama değeri eklenerek 3 byte'lık *RV3* değeri elde edilir. Doğrulama değeri özellikle kesme saldırıları ve gürültü saldırılarında *R2* değerinin sağlık bir şekilde elde edilebilmesi için elzemdir ve şifrelenmiş matris içerisinde asıl aklanacak olan bu *RV3* değeridir.

Bu şekilde hesaplanan *RV3* veri bloğunu şifrelenmiş resim dosyasına dağıtmak için öncelikle resim dosyasının sonuna yeni bir satır eklenir. Bu boş satırın tamamını dolduracak şekilde *RV3* değeri tekrarlanarak yazılır. Matrisin sütun uzunluğu *90* olduğunu düşünelim, bu durumda *30* kez *(90 / 3 = 30) RV3* değeri art arda yeni eklenen satıra yazılacaktır. Daha sonra birinci tent haritadan elde edilen *m* adet veri ile yeni eklenen satır XoR işlemine tabi tutulur. Bu şekilde *RV3* değerlerinin güvenliği sağlanmış olur. Son aşamada ise yine birinci tent haritadan *m/3* adet ikili değerler elde edilir. Bu ikili değerler değiş-tokuş (swap) işleminde kullanılacak matris içerisindeki indisleri ifade etmektedir ve bu değerlere göre son satırda bulunan şifrelenmiş veri yine şifrelenmiş matris içerisine dağıtılır. Bu son işlemin çıktısı ile nihai şifrelenmiş dosya elde edilmektedir.



a) orijinal dosya b) şifreli dosya R2 (1_237) c) şifreli dosya R2 (127_26)

d) şifreli dosya R2 (170_57) e) şifreli dosya R2 (230_49) f) şifreli dosya R2 (255_238)

Şekil 3.11: Olasılıksal şifreleme örnekleri.

a) Orijinal resim dosyası, b) orijinal dosyanın *R2* (1, 237) ile şifrelenmesi, c) orijinal dosyanın *R2* (127, 26) ile şifrelenmesi, d) orijinal dosyanın *R2* (170, 57) ile şifrelenmesi, e) orijinal dosyanın *R2* (230, 49) ile şifrelenmesi, f) orijinal dosyanın *R2* (255, 238) ile şifrelenmesi.

Şekil 3.11' de aynı dosyanın aynı gizli anahtar ile art arda 5 kez şifrelenmesi sonucu elde edilen şifrelenmiş dosyalar verilmiştir. Her bir şifreleme adımında üretilen ve ikinci kaotik haritanın başlangıç parametrelerinde kullanılan *R2* değerleri her bir dosyanın yanında belirtilmiştir. Tablo 3.3' de ise elde edilen şifrelenmiş dosyaların benzerlikleri PSNR değerleri üzerinden yapılmıştır. Her bir karşılaştırmada elde edilen PSNR değerlerine bakılacak olursa aynı dosyanın aynı anahtar ile şifrelenmesi sonucunda elde edilen şifrelenmiş dosyaların birbirlerinden oldukça farklı olduğu görülecektir ki bu durum geliştirmiş olduğumuz olasılıksal şifreleme algoritmasının oldukça başarılı sonuçlar ürettiğini göstermektedir.

Tablo 3.3: Şifreli dosyaların benzerlik (PSNR) karşılaştırması.

	C. (1_237)	**C. (127_26)**	**C. (170_57)**	**C. (230_49)**	**C. (255_238)**
C. (1_237)	∞	27.684858	27.843302	27.192278	27.356892
C. (127_26)		∞	27.124885	27.285163	27.834698
C. (170_57)			∞	27.344674	27.457355
C. (230_49)				∞	27.301913
C. (255_238)					∞

3.4.2. Şifre Çözme Adımları

Şifre çözme (decryption) aşamasında şifreleme aşamasında uygulanan adımlar ters sırada tek tek uygulanmıştır. Öncelikle şifreli resim dosyasından 2 byte'lık *R2* değerinin elde edilmesi gerekmektedir. Bunun için yalnızca gizli anahtar kullanılarak başlatılan birinci tent haritadan elde edilen rastgele veriler ile şifrelenmiş resim dosyası içerisine şifreli ve rastgele bir şekilde dağıtılmış olan 3 byte'lık veri blokları (*RV3*) resim dosyasının en alt satırında toplanmıştır. Daha sonra yine birinci tent haritadan elde edilen rastgele veriler ile resmin son satırına XoR işlemi uygulanarak şifreli değerler çözülmüştür.

Son satırda toplanan ve tekrarlanan *RV3* veri bloğundan *R2* değerinin elde edilmesi gerekmektedir. Bunun için öncelikli olarak son satır verileri 3 byte'lık bloklar şeklinde parçalara ayrılmıştır. Her bir parça için doğrula işlemi uygulanmıştır ve doğrulama işleminden geçen *RV3* değerlerine ait son byte iptal edilerek 2 byte'lık bloklar şeklinde alınıp kullanım

frekanslarına bakılmıştır. Kullanım frekansı en yüksek olan 2 byte'lık veri bloğu *R2* değeri olarak belirlenmiştir. Bir sonraki aşamada ise buradan elde edilen bu 2 byte'lık *R2* değeri gizli anahtar ile birlikte kullanılarak ikinci kaotik haritanın başlangıç parametrelerinin belirlenmiştir. *R2* değeri belirlendikten sonra resim dosyasına ait son satır silinmiş ve orijinal şifrelenmiş dosya elde edilmiştir. İkinci tent haritadan *n x m* boyutlarındaki rastgele değerlerden oluşan bir matris üretilmiş ve bu matris ile şifrelenmiş veri XoR işlemine tabi tutulmuştur.

Son aşamada XoR işlemi ile çözülen veriye Knuth – Durstenfeld karıştırma algoritması ters yönlü olarak uygulanmıştır. Burada yine birinci kaotik haritadan elde edilen rastgele değerler kullanılmıştır. Bu işlemden sonra orijinal dosya elde edilmektedir.

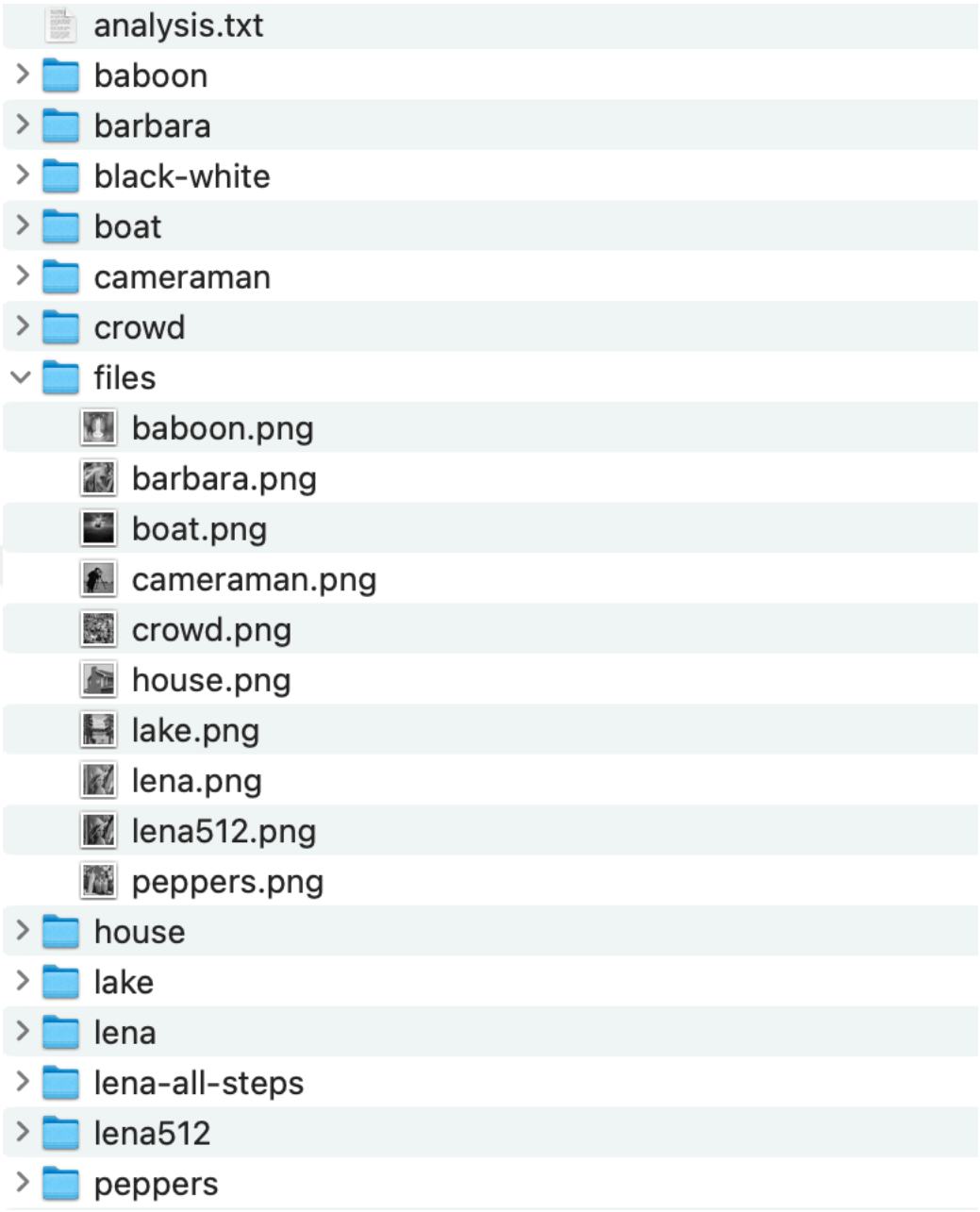
4. BULGULAR

Bu bölümde tez çalışması kapsamında geliştirilen şifreleme ve şifre çözme algoritmaları farklı resim dosyaları üzerinde detaylı olarak test edilmiştir. Benzer çalışmalarda kullanılan *256x256* gri tonlamalı PNG formatında; "lena.png", "cameraman.png", "baboon.png", "peppers.png", "barbara.png", "boat.png", "lake.png" ve "hause.png" olmak üzere 8 ayrı resim dosyası kullanılmıştır. Ayrıca diğer çalışmalar ile karşılaştırmak için *256x256* gri tonlamalı "crowd.png" ve *512x512* gri tonlamalı "lena.png" resim dosyaları kullanılmıştır. Yine aynı şekilde tamamen siyah ve tamamen beyaz renklerden oluşan gri tonlamalı resim dosyaları da test için kullanılmıştır.

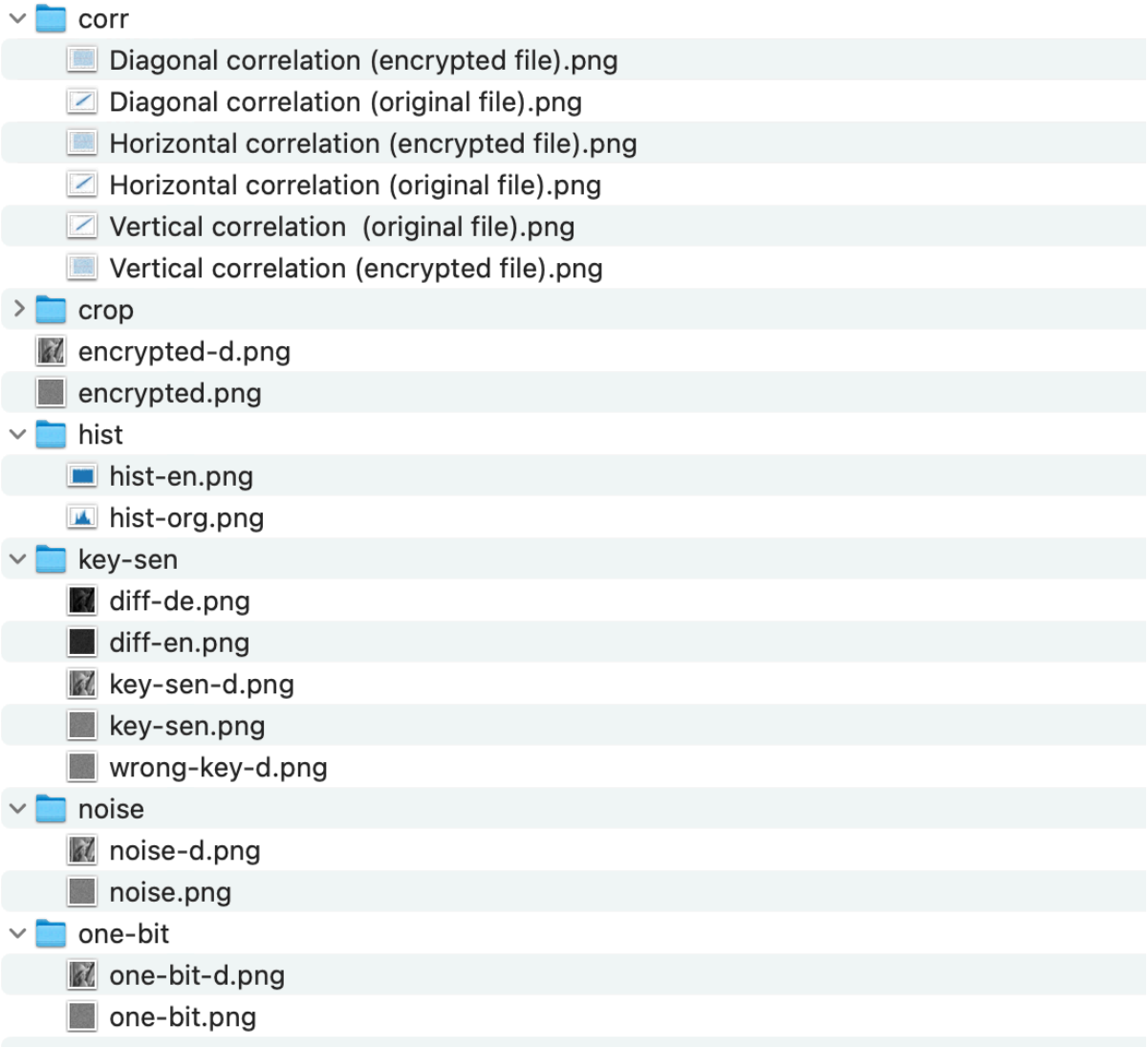
Başarılı bir şifreleme algoritmasın geçmesi gereken zorunlu bazı test ve analiz adımları şu şekildedir: Anahtar Analizi (Key Analysis), Bilgi Entropi Analizi (Information Entropy Analysis), İstatistiksel Saldırı Analizi (Statistical Attack Analysis), Diferansiyel Saldırı Analizi (Differential Attack Analysis) ve Dayanıklılık Analizleridir (Robustness Analysis). Bu bölümde bütün bu test ve analiz adımları 8 ayrı resim dosyası için tek tek uygulanmış ve sonuçları detaylı bir şekilde paylaşılmıştır. Şifreleme, şifre çözme ve test/analiz kodları Python 3.9.2 programlama dili kullanılarak Visual Studio Code editöründe geliştirilmiştir. Bütün testler i7 Intel işlemcili 8 GB hafızaya sahip Mac OS X bilgisayar ortamında yapılmıştır.

Test ortamı olarak hazırlanan dizin yapısı Şekil 4.1'de gösterilmiştir. Test işlemine tabi olacak bütün dosyalar "files" dizininde tutulmaktadır. Test aşaması başlatılmadan önce "initializeFolderStructure" modülü çalıştırılarak "files" dizininde bulunan her bir dosya için Şekil 4.2'de detayı verilen dizin yapısı oluşturulmaktadır. Daha sonra çalıştırılan "testEncryption", "testDecryption" ve "runAnalysis" modülleri ile "files" dizininde bulunan her bir dosya için Şekil 4.2'de detayları verilen dosyalar ilgili alt dizinlerde otomatik olarak oluşturulmaktadır. Ayrıca "black-white" dizininde tamamen siyah ve tamamen beyaz renklerden oluşan dosyalar için gerekli test çıktıları oluşturulmaktadır. Yine "lena-all-steps" dizininde "Lena" dosyasına özgü olarak şifreleme ve şifre çözme adımlarında elde edilen dosyalar ayrı ayrı tutulmuştur.

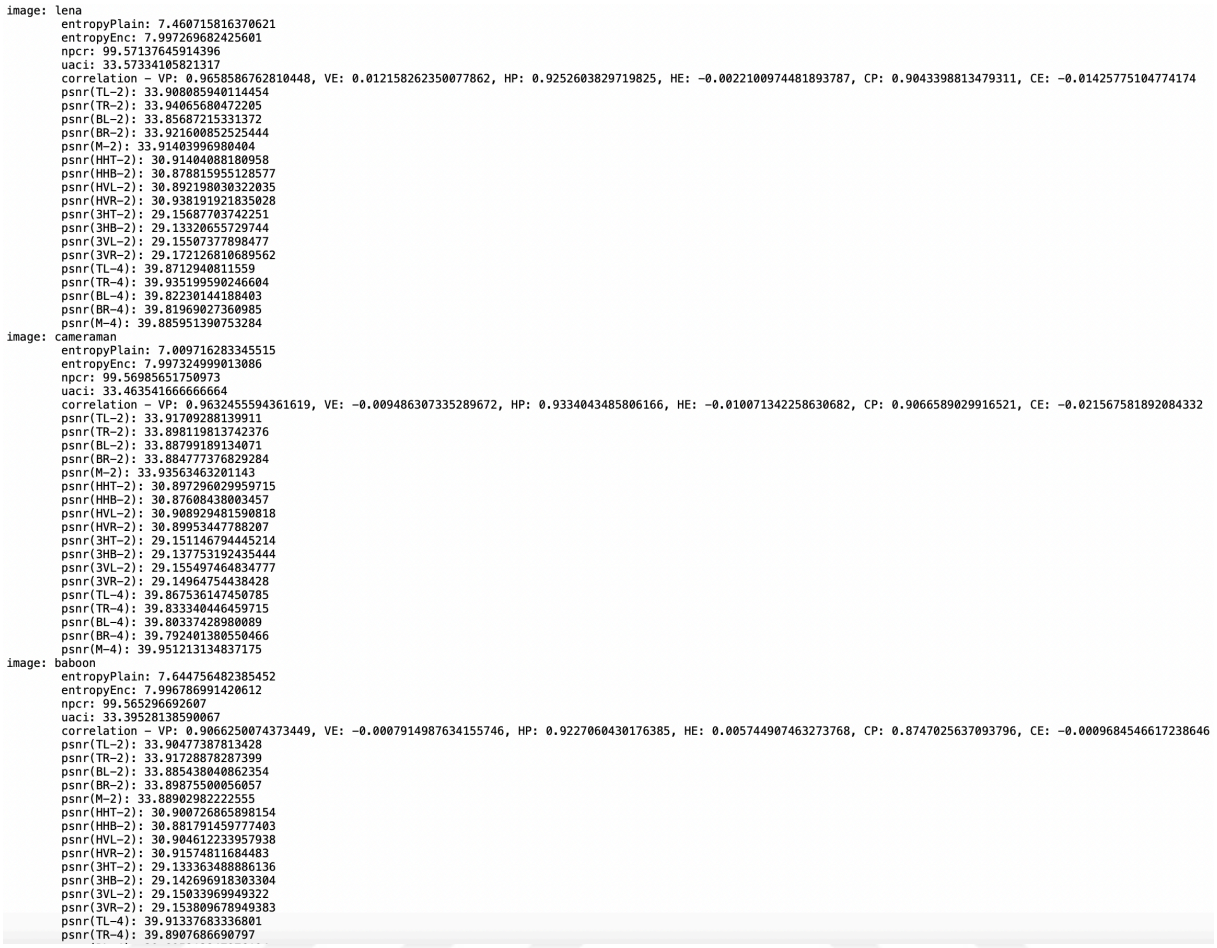
Son olarak "runAnalysis" modülü çalıştırıldığında her bir dosya için elde edilen entropi, NPCR, UACI, korelasyon katsayıları (yatay, dikey, çapraz), PSNR değerleri hesaplanmış ve Şekil 4.3'de gösterildiği gibi formatlı bir şekilde "analysis.txt" isimli dosyaya yazılmıştır.



Şekil 4.1: Test ortamı dizin yapısı.



Şekil 4.2: "Lena" dosyası için oluşturulan alt dizinler ve dosyalar.



Şekil 4.3: Her bir dosya için hesaplanan analiz sonuçları.

4.1. ANAHTAR ANALİZİ (KEY ANALYSIS)

Anahtar analizinde iki farklı analiz yapılmaktadır, bunlar; Anahtar Uzayı Analizi (Key Space Analysis) ve Anahtar Hassasiyet Analizi (Key Sensitivity Analysis)'dir. Anahtar uzayı analizinde anahtar uzunluğu ve kaba saldırılara karşı algoritmanın dayanıklılığı analiz edilirken anahtar hassasiyet analizinde ise şifrelenmiş dosyanın şifrelemede kullanılan gizli anahtara olan bağlılığı analiz edilmektedir.

4.1.1. Anahtar Uzayı Analizi (Key Space Analysis)

Anahtar uzayı analizi algoritmanın kaba kuvvet saldırılarına karşı dayanıklılığını test etmek için kullanılan bir analiz çeşididir. Günümüz bilgisayarları düşünüldüğünde bir algoritmanın başarılı sayılabilmesi için anahtar uzunluğunun minimum 2^{100} yani 100 bit olması gerekmektedir.

Algoritma kapsamında veri karıştırma ve XoR işlemleri esnasında iki farklı lojistik harita kullanıldığı için 4 farklı başlangıç parametresi vardır. Birinci kaotik harita başlangıç parametreleri; L_{x0} ve L_r ve ikinci kaotik harita başlangıç parametreleri; S_{x0} ve S_r olarak sembolize edilmiştir. 64 bit işlem kabiliyeti olan bir bilgisayar için rasyonel sayı hassasiyeti (double-precision) 10^{15} dir. Lojistik harita başlangıç parametresi olan x_0 10^{15} aralığında herhangi bir değer alabilirken r kontrol parametresi $0{,}43\ x\ 10^{15}$ aralığında herhangi bir değer alabilir [97]. Bu durumda geliştirmiş olduğumuz şifreleme algoritması için anahtar uzunluğu Formül (4.1)'de verildiği gibi hesaplanır ve yaklaşık olarak 2^{196} yani *196* bit olarak bulunur.

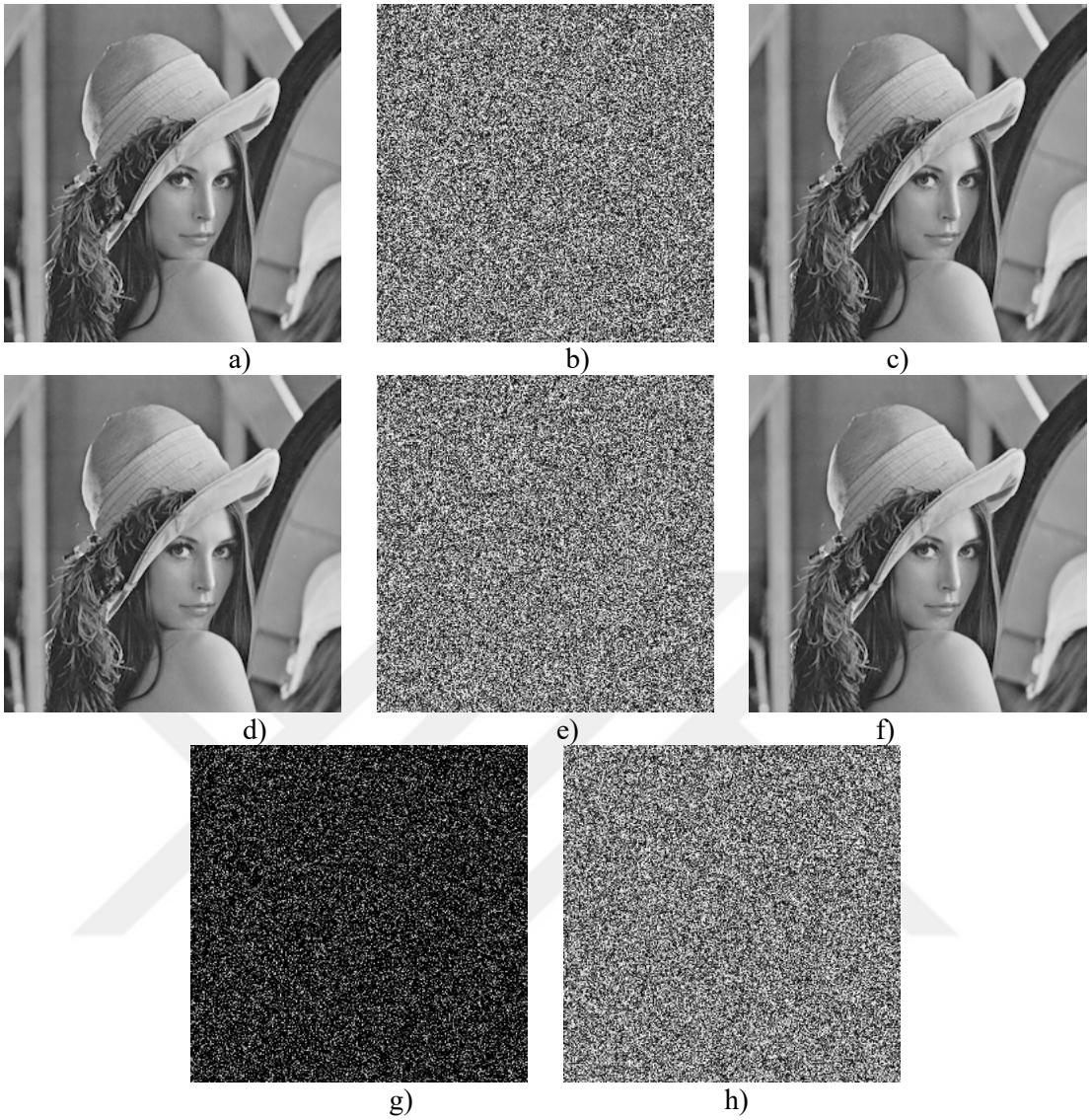
$$0.43 * 10^{15} * 10^{15} * 0.43 * 10^{15} * 10^{15} \approx 2^{196} \tag{4.1}$$

196 bit anahtar uzunluğu algoritmanın yeterli düzeyde güvenli olduğunu ve günümüz modern bilgisayarları tarafında yapılacak kaba kuvvet saldırılarına karşı dirençli olduğunu göstermektedir.

4.1.2. Anahtar Hassasiyet Analizi (Key Sensitivity Analysis)

Anahtar hassasiyet analizi bir şifreleme algoritmasının anahtara olan bağlılığını ölçmek için kullanılan bir analiz çeşididir. Başarılı bir şifreleme algoritmasından şifrelemede kullanılan anahtara karşı çok hassas olması beklenir. Anahtar üzerinde yapılacak çok küçük bir değişikliğin şifrelenmiş veri üzerinde çok büyük etki oluşturması ve bu sayede algoritmanın bu tür saldırılara karşı dirençli olması beklenir.

Bu amaçla yapılan testlerde şu adımlar takip edilmiştir; birinci aşamada resim dosyası *K1* anahtarı ile şifrelenmiştir. İkinci aşamada *K1* anahtarına 10^{-15} gibi çok küçük bir değer eklenerek *K2* anahtarı elde edilmiştir. Üçüncü aşamada orijinal resim dosyası bu kez *K2* anahtarı kullanılarak şifrelenmiştir. Dördüncü aşamada şifrelenmiş her bir resim dosyası kendi anahtarları kullanılarak açılmıştır. Beşinci aşamada *K1* anahtarı ile şifrelenerek elde edilen resim dosyası ile *K2* anahtarı kullanılarak elde edilen şifrelenmiş resim dosyaları arasındaki fark incelenmiştir. Son aşamada ise *K1* anahtarı kullanılarak şifrelenmiş olan resim dosyası *K2* anahtarı kullanılarak açılmıştır. Şekil 4.4'de anahtar hassasiyet analizi test sonuçları gösterilmiştir.



Şekil 4.4: Anahtar hassasiyet analizi sonuçları.

a) orijinal resim, b) *K1* anahtarı ile şifrelenmiş resim, c) *K1* anahtarı ile açılmış resim, d) orijinal resim, e) *K2 (K2 = K1 +* $\mathbf{10^{-15}}$*)* anahtarı ile şifrelenmiş resim, f) *K2* anahtarı ile açılmış resim, g) b ve e resim dosyaları farkı, h) *K1* anahtarı ile şifrelenmiş resim dosyasının *K2* anahtarı ile açılması.

4.2. BİLGİ ENTROPİ ANALİZİ (INFORMATION ENTROPY ANALYSIS)

Bir algoritmanın bilgi entropisi saldırılarına karşı dayanıklı olabilmesi için bilgi entropi değerine bakılır. Bilgi entropisi, uzay kümesinin yani piksel değerlerinin ne kadar dengeli kullanıldığını, bir bölgede yığılma olup olmadığını ölçmek için kullanılır ve Formül (4.2)'de verildiği şekilde hesaplanır.

$$H(X) = \sum_{i=1}^{n} P_r(x_i) \log_2 \frac{1}{P_r(x_i)} \tag{4.2}$$

Burada X değeri bilgi entropisi hesaplanacak olan resim dosyasını, x_i resim dosyası içerisindeki herbir pikselin değerini, $P_r(x_i)$ her bir piksel değerinin olasılığını gösterir. Uzay kümesi 2^n olan bir veri seti için ideal entori değeri *n*'dir. Testlerde kullandığımız gri tonlamalı örnek resim dosyaları *256* farklı renk içerebilmektedir ve $2^8 = 256$ olduğundan dolayı örnek resim dosyaları için ideal entropi değeri *8*'dir. Şifrelenmiş resim dosyası entropi değeri ne kadar çok *8* değerine yakınsarsa algoritma bilgi entropisi açısından o kadar güçlüdür, *8* değerinden ne kadar uzaksarsa algoritmanın bilgi entropisi saldırılarına olan dayanıklılığı o kadar zayıftır denilebilir.

Tablo 4.1'de geliştirilen şifreleme algoritmasının örnek resim dosyaları üzerindeki bilgi entropi değerleri bulunmaktadır. Burada hem orijinal resim dosyalarının bilgi entropi değerleri hesaplanmış hem de şifrelenmiş resim dosyasının bilgi entropi değerleri hesaplanmıştır. Alanda yapılan çalışmalara göre *256* gri tonlamalı resim dosyaları için başarılı bir şifreleme algoritmasının bilgi entropi değerinin asgari *7.99* olması beklenmektedir. Bu değerin altında bilgi entropisi değerine sahip şifreleme algoritmaları bilgi entropi saldırısı açısından kırılgan olarak kabul edilmektedir. Tablo 4.1'de sunulan değerleri incelediğimizde *8* farklı şifrelenmiş resim dosyaları için üretilen bilgi entropi değeri *7.9945*'in üzerinde olup geliştirilen şifreleme algoritmasının bilgi entropi testinden başarılı bir şekilde geçtiğini söyleyebiliriz.

Tablo 4.1: Şifreli ve açık resim dosyalarının bilgi entropi değerleri.

Resim Dosyası	**Boyut**	**Bilgi Entropisi (Açık)**	**Bilgi Entropisi (Şifreli)**
Lena	256x256	7.460715	7.996965
Baboon	256x256	7.644756	7.997000
Cameraman	256x256	7.009716	7.997554
Peppers	256x256	7.688517	7.997168
Barbara	256x256	7.599517	7.997031
Boat	256x256	7.301134	7.997265
Lake	256x256	7.488187	7.996776
House	256x256	7.577239	7.994579

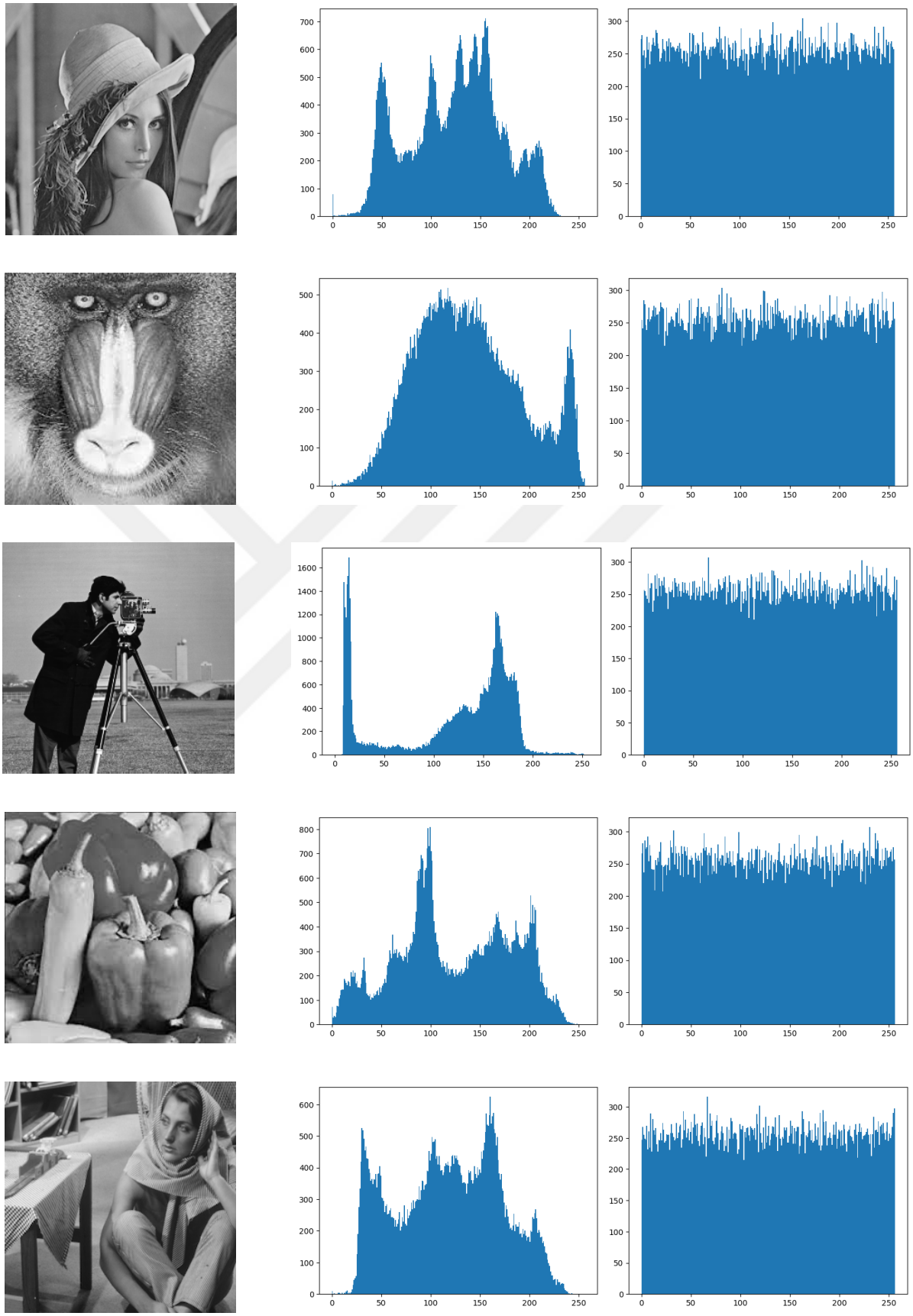
4.3. İSTATİSTİKSEL SALDIRI ANALİZİ (STATISTICAL ATTACK ANALYSIS)

İstatistiksel saldırı analizi, Histogram Analizi (Histogram Analysis) ve Korelasyon Analizi (Correlation Analysis) olmak üzere iki alt başlıkta incelenmiştir. Histogram analizinde açık ve şifrelenmiş resim dosyalarının histogramları çıkartılıp analiz yapılmaktadır. Korelasyon analizinde ise açık ve şifrelenmiş resim dosyalarındaki piksellerin yatay, dikey ve çapraz komşuluklarına bakılarak birbirleriyle olan korelasyonlar analiz edilmektedir. Testlerde her bir komşuluk için korelasyon katsayıları ayrı ayrı hesaplanıp incelenmektedir.

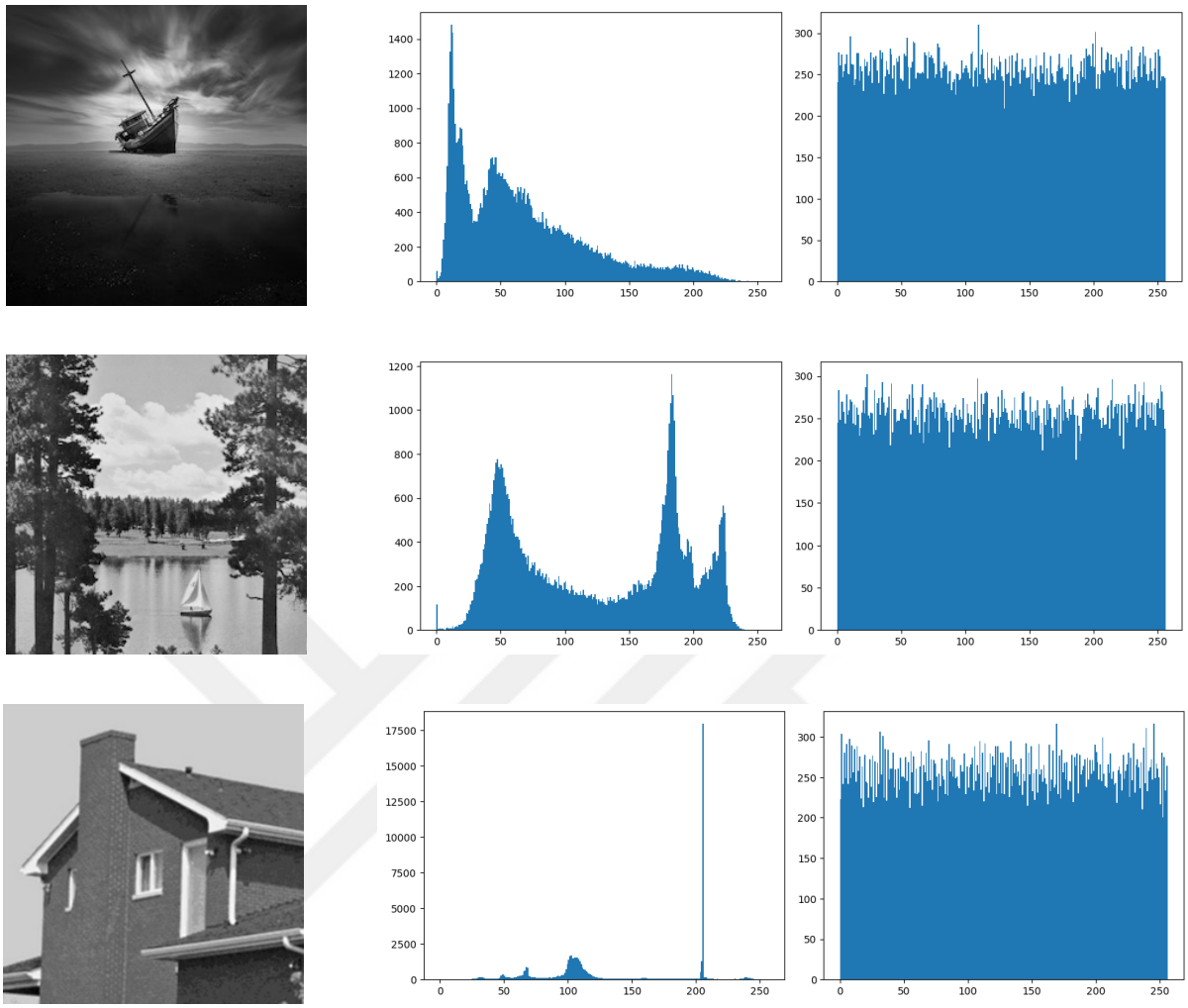
4.3.1. Histogram Analizi (Histogram Analysis)

Histogram analizinde bir resim dosyasında kullanılan renklerin ne yoğunlukta kullanıldıklarının dağılımına bakılmaktadır. Normal bir resim dosyasında renk kullanımları dengeli olmayıp bazı renklerin kullanımı diğer renklere göre daha fazla, bazılarına göre daha azdır. Bu durum şifre kırıcılara (kriptanalist) resim dosyası hakkında çeşitli ipuçları vermektedir. Başarılı bir şifreleme algoritmasının orijinal resim dosyası hakkında hiçbir veri sızdırmaması beklenmektedir. Bundan dolayı şifrelenmiş resim dosyalarının histogramlarında bütün renklerin dengeli bir şekilde kullanılması gerekmektedir ki bu sayede kriptanalistlerin saldırıları engellenebilmiş olsun.

Şekil 4.5'de test amaçlı kullanılan *8* adet gri tonlamalı resim doyasına ait histogram analizi listelenmiştir. Birinci kolonda orijinal resim dosyası, ikinci kolonda orijinal resim dosyasına ait histogram çıktısı, üçüncü kolonda ise şifrelenmiş resim dosyasına ait histogram çıktısı verilmiştir. Orijinal resim dosyalarına ait histogramlarda beklendiği gibi renk kullanım yoğunlukları dengesiz bir şekilde dağılmıştır. Şifrelenmiş resim dosyalarında ise renk kullanım sıklıkları birbirlerine çok yakın seyretmiştir. Bu durum zaten başarılı bir şifreleme algoritmasından beklenen çıktı olup geliştirilen algoritmanın histogram analizinden başarılı bir şekilde geçtiğini göstermektedir.



Şekil 4.5: Histogram analizi.



Şekil 4.5 (devam): Histogram analizi.

4.3.2. Korelasyon Analizi (Correlation Analysis)

Resim dosyalarında komşu pikseller arasındaki ilişki oldukça yükseltir. Bir pikselin sağındaki, solundaki, yukarısındaki, aşağısındaki ya da çaprazındaki pikseller ile olan ilişkisi oldukça yüksektir. Yani komşu pikseller arasındaki renk değişimleri genellikle çok azdır. Bu durum resim dosyası şifreleme işlemini diğer dosya türlerini şifrelemeye göre daha zorlaştırmaktadır. Bu yüzden şifreleme algoritmalarının başarı analizleri genellikle resim dosyaları üzerinden yapılmaktadır. Başarılı bir şifreleme algoritmasının orijinal resim dosyası hakkında hiçbir bilgi sızdırmaması beklenir. Bunun için başarılı bir şifreleme algoritması komşu pikseller arasındaki bu korelasyonu bozmalı ve saldırganlara orijinal dosya hakkında hiçbir ipucu vermemelidir.

Pikseller arasındaki korelasyonu belirlemek için korelasyon katsayısına (correlation coefficient) bakılır. Korelasyon katsayısı hesaplaması, Formül (4.3), Formül (4.4), Formül (4.5) ve Formül (4.6)'da verilmiştir. Buradaki her bir x ve y değeri resim dosyası içerisinde geçen komşu iki pikselin renk değerini göstermektedir.

$$r_{xy} = \frac{cov(x,y)}{\sqrt{D(x)}\sqrt{D(y)}} \tag{4.3}$$

$$cov(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)) \tag{4.4}$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2 \tag{4.5}$$

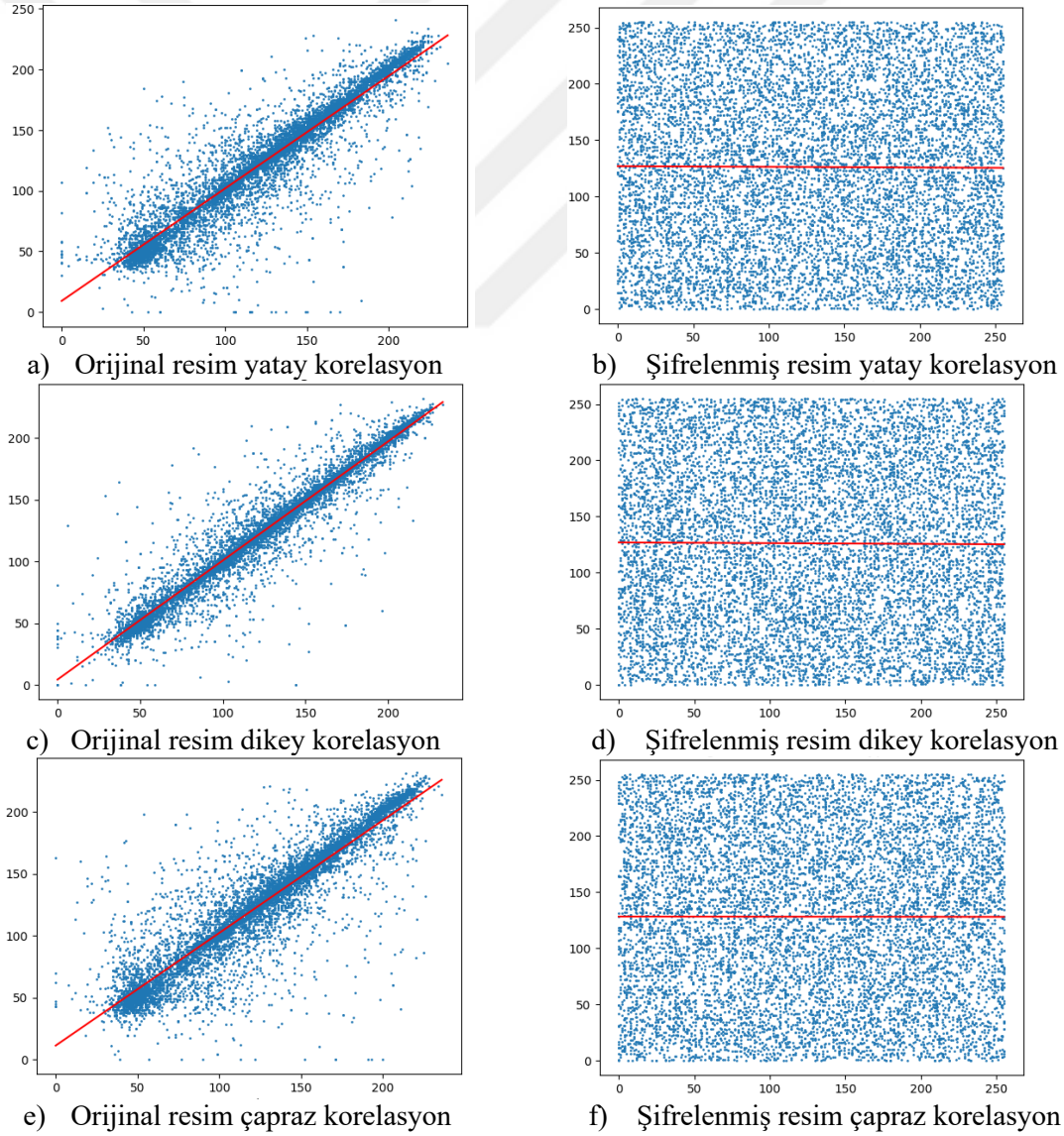
$$E(x) = \frac{1}{N}\sum_{i=1}^{N}x_i \tag{4.6}$$

Korelasyon analizinde genellikle üç farklı korelasyona bakılır, bunlar; yatay korelasyon (horizontal correlation), dikey korelasyon (vertical correlation) ve çapraz korelasyondur (diagonal correlation). Başarılı bir şifreleme algoritmasından bu üç farklı korelasyonu tamamen bozması ve orijinal dosya hakkında hiçbir ipucu sızdırmaması beklenmektedir. Geliştirilmiş olan şifreleme algoritmasının korelasyon testleri için *8* ayrı resim dosyasında bu *3* farklı korelasyon değerleri analiz edilmiştir. Bu kapsamda her bir şifrelenmiş ve açık resim dosyalarından *10000*'er adet komşu piksel değerleri rastgele seçilerek aralarındaki ilişki hesaplanmıştır.

Şekil 4.6'da "Lena" resim dosyası için yapılan korelasyon analizi verilmiştir. Yatay korelasyon, dikey korelasyon ve çapraz korelasyonlar ayrı ayrı incelenmiştir. Bu 3 ayrı korelasyon analizi için hem açık resim dosyasından hem de şifrelenmiş resim dosyasından *10000* adet komşu pikseller alınmıştır ve pikseller arasındaki ilişki hesaplanmıştır. a) ile

belirtilen şekilde açık resim dosyasının yatay korelasyon analizi sunulmuştur. Aynı analizin şifrelenmiş dosya üzerinden elde edilen çıktısı b) ile belirtilen grafikte sunulmuştur. Açık resim dosyası için dikey korelasyon c)'de sunulmuştur. Şifreli resim dosyası için dikey korelasyon d)'de sunulmuştur. Çapraz korelasyon için e) ve f) grafikeri verilmiştir, e)'de açık resim dosyasının çapraz korelasyonu verilmişken f)'de şifreli resim dosyası için çapraz korelasyon grafiği sunulmuştur.

Şekil 4.6'da sunulan grafiklerden de rahatlıkla görülebileceği gibi açık resim dosyalarında komşu pikseller arasında sıkı bir korelasyon varken şifrelenmiş resim dosyalarında bu korelasyon tamamen bozulmuştur. Bu sayede kriptanalistlerin şifrelenmiş resim dosyası üzerinden orijinal dosya hakkında bilgi edinebilmesinin önene geçilmiştir.



a) Orijinal resim yatay korelasyon
b) Şifrelenmiş resim yatay korelasyon
c) Orijinal resim dikey korelasyon
d) Şifrelenmiş resim dikey korelasyon
e) Orijinal resim çapraz korelasyon
f) Şifrelenmiş resim çapraz korelasyon

Şekil 4.6: Lena resim dosyası için korelasyon analizi.

Tablo 4.2'de test amaçlı kullanılan *8* ayrı resim dosyasının hem açık hem de şifreli durumları için yatay, dikey ve çapraz korelasyon katsayıları hesaplanmıştır. Tablo dikkatli incelendiğinde açık resim dosyaları için ortalama *0.95* civarında olan korelasyon katsayı değerleri şifrelenmiş resim dosyaları için ortalama *0.003* seviyelerine düşmüştür. Bu da şifreleme algoritmasının oldukça başarılı sonuçlar ürettiğini göstermektedir.

Tablo 4.2: Şifreli ve açık resim dosyaları için hesaplan korelasyon katsayıları.

Dosya Adı	Açık Resim Dosyası			Şifreli Resim Dosyası		
	Yatay	Dikey	Çapraz	Yatay	Dikey	Çapraz
Lena	0.925517	0.962879	0.908886	-0.005877	-0.006477	0.001014
Baboon	0.922564	0.901886	0.874602	0.003462	-0.003645	0.019347
Cameraman	0.939263	0.961445	0.910596	0.006066	0.010519	0.009647
Peppers	0.965953	0.971333	0.939607	0.008350	-0.019804	0.003081
Barbara	0.932332	0.954344	0.913083	0.000103	0.000857	0.026793
Boat	0.983629	0.977808	0.968991	0.000651	-0.023207	-0.013492
Lake	0.947915	0.946228	0.916189	0.001175	0.013189	-0.009658
House	0.983292	0.980817	0.966862	0.004782	0.006357	0.018437

4.4. DİFERANSİYEL SALDIRI ANALİZİ (DIFFERENTIAL ATTACK ANALYSIS)

Kriptanalistler şifrelenmiş veriden orijinal veriye ulaşmak için birçok teknik kullanmaktadırlar. Bunlardan yayın olarak kullanılan metotlardan birisi de diferansiyel saldırılardır. Bu saldırı türünde orijinal veri bir gizli anahtar kullanılarak normal bir şekilde şifrelenir ve şifrelenmiş veri elde edilir. Daha sonra orijinal veride 1 bitlik bir değişiklik yapılır ve değiştirilmiş yeni veri yine aynı gizli anahtar kullanılarak şifrelenir. Elde edilen iki şifrelenmiş dosya üzerinde detaylı incelemeler yapılarak orijinal veri hakkında bilgi edinilmeye çalışılır.

Bir şifreleme algoritmasının diferansiyel saldırılara karşı dayanıklılığını test etmek için iki ayrı parametreye bakılır, bunlar; değişikliğe uğramış piksel oranı (Number of Pixels Change Rate – NPCR) ve birleşik ortalama değişim yoğunluğudur. (Unified Average Changing Intensity – UACI). Formül (4.7) ve Formül (4.8)'de NPCR değerinin nasıl hesaplandığı verilmiştir. Formül (4.9)'da ise UACI değerinin nasıl hesaplandığı sunulmuştur.

$$NPCR = \frac{\sum_{i,j} S(x,y)}{N * M} * 100\% \tag{4.7}$$

Burada *M* ve *N* değerleri resim dosyasının boyutlarını göstermektedir. *S*(*x*, *y*) Formül (4.8)'de sunulduğu gibi hesaplanmaktadır.

$$S(x,y) = \begin{cases} 1, & if\ C1(x,y) \neq C2(x,y) \\ 0, & if\ C1(x,y) = C2(x,y) \end{cases} \tag{4.8}$$

Burada *C1* ilk şifrelenmiş resim dosyasını göstermekte *C2* ise orijinal resim dosyasında bir bitlik değişiklik yapıldıktan sonra elde edilen ikinci şifrelenmiş resim dosyasını ifade etmektedir.

$$UACI = \frac{1}{N * M}\left(\sum_{i,j} \frac{|C1(i,j) - C2(i,j)|}{255}\right) * 100\% \tag{4.9}$$

Tablo 4.3'de *8* farklı test dosyası için hesaplan NPCR ve UACI değerleri sunulmuştur. Mevcut çalışmalara bakıldığında, bir şifreleme algoritmasının diferansiyel saldırılara karşı dayanıklı olduğunu söyleyebilmemiz için NPCR değerinin *99%*'den büyük olması gerekirken UACI değerinin *33%* değerine yakınsaması gerekmektedir.

Tabloyu detaylı bir şekilde incelediğimizde test için kullanılan *8* farklı resim dosyası için elde edilen NPCR ve UACI değerlinin oldukça tatmin edici sonuçlar olduğunu rahatlıkla söyleyebiliriz. Yani bu sonuçlardan yola çıkarak, önerilen şifreleme algoritması diferansiyel saldırılara karşı dayanıklı bir algoritmadır diyebiliriz.

Tablo 4.3: Test dosyaları için elde edilen NPCR ve UACI değerleri.

Dosya Adı	**Boyut**	**NPCR(%)**	**UACI(%)**
Lena	256x256	99.577456	33.411231
Baboon	256x256	99.585055	33.453283
Cameraman	256x256	99.542497	33.495001
Peppers	256x256	99.589615	33.449057
Barbara	256x256	99.574416	33.528249
Boat	256x256	99.585055	33.579420
Lake	256x256	99.563776	33.431879
House	256x256	99.588095	33.520518

4.5. DAYANIKLILIK ANALİZİ (ROBUSTNESS ANALYSIS)

Şifrelenmiş dosyalar saklandığı ortamda yaşanan dijital sorunlar nedeniyle ya da bir sistemden diğer sisteme aktarım esnasında ağda yaşanabilecek sorunlar nedeniyle veri kaybına uğrayabilir. Bu kayıplar bazen küçük olabileceği gibi bazı durumlarda ciddi oranlara ulaşabilir. Başarılı bir şifreleme algoritmasının bu tür veri kayıplarına karışı dirençli olması, kayıplara rağmen orijinal veriyi elde edebiliyor olması beklenir.

İki resim dosyası arasındaki benzerliği ölçmek için En Yüksek Sinyal-Gürültü Oranı (Peak Signal-to-Noise Ratio – PSNR) değerine bakılır. Bu değer ne kadar yüksek ise iki dosya arasındaki benzerlik o kadar yüksektir, ne kadar düşük ise dosyalar arasındaki benzerlik o kadar düşüktür denir. Aynı dosyalar için PSNR değeri sonsuz olarak hesaplanır. PSNR değeri Formül (4.10)'da gösterildiği gibi hesaplanır.

$$PSNR = 10 * \log_{10}\left(\frac{255 * 255}{MSE}\right)(dB) \tag{4.10}$$

Burada kullanılan MSE (Mean Square Error – Ortalama Karesel Hata) değeridir ve Formül (4.11)'de gösterildiği gibi hesaplanmaktadır.

$$MSE = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\|I_1(i,j) - I_2(i,j)\|^2 \tag{4.11}$$

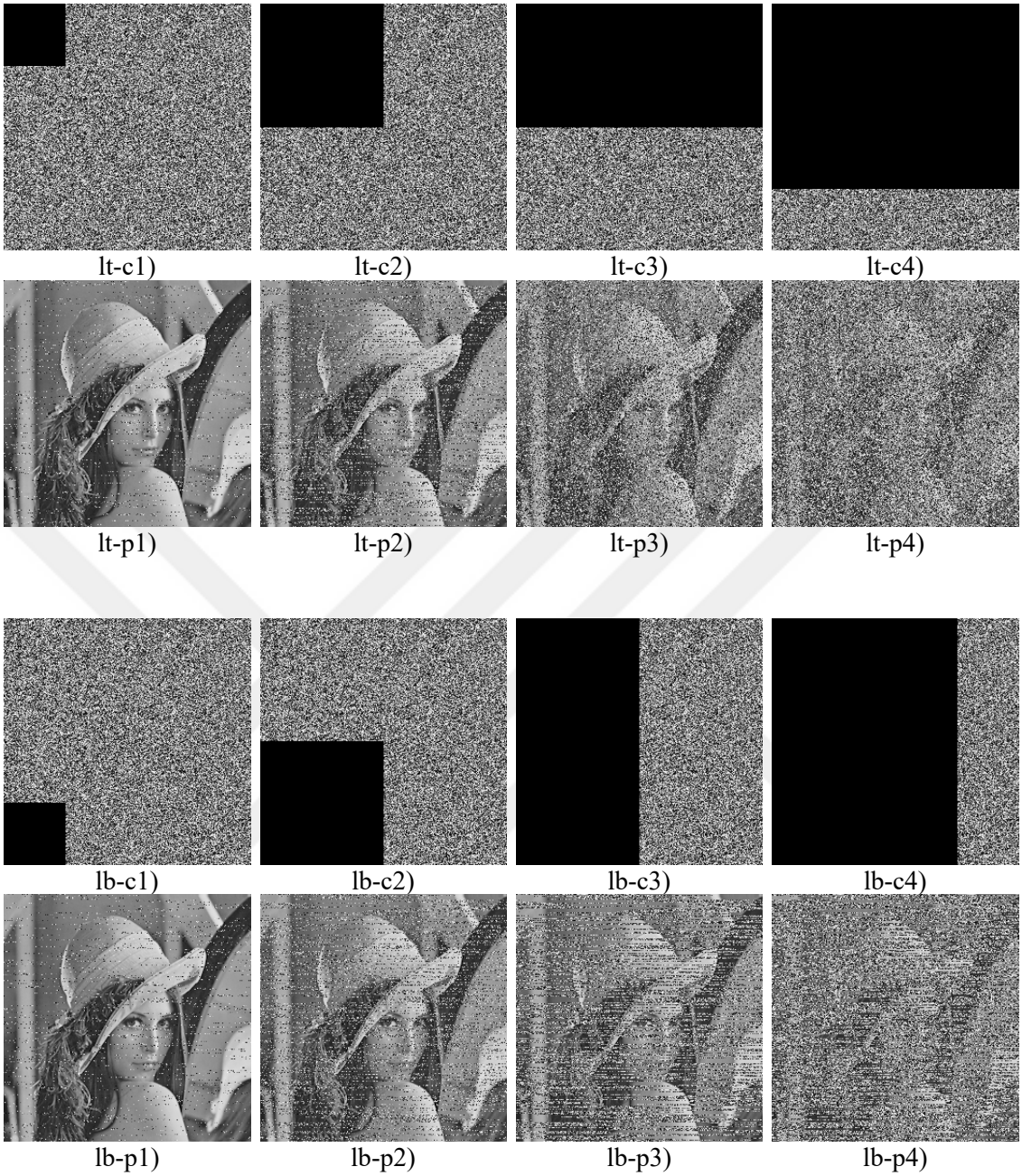
Burada kullanılan *n* resim dosyasının satır sayısı, *m* ise sütun sayısını göstermektedir. Dayanıklılık analizi, Kırpma Saldırısı (Cropping Attack) ve Gürültü Saldırısı (Noise Attack) olmak üzere iki farklı şekilde incelenmektedir. Kırpma saldırısında şifrelenmiş resim dosyalarının farklı bölgelerinden farklı büyüklerde veri silinir, gürültü saldırısında ise şifrelenmiş verinin rastgele seçilen noktalarına saldırı yapılır. Her iki saldırı şeklinde de orijinal dosya ile saldırıya uğramış yani veri kaybı yaşamış dosya PSNR değeri hesaplanır ve bu değerin yüksek çıkması amaçlanır.

4.5.1. Kırpma Saldırısı (Cropping Attack)

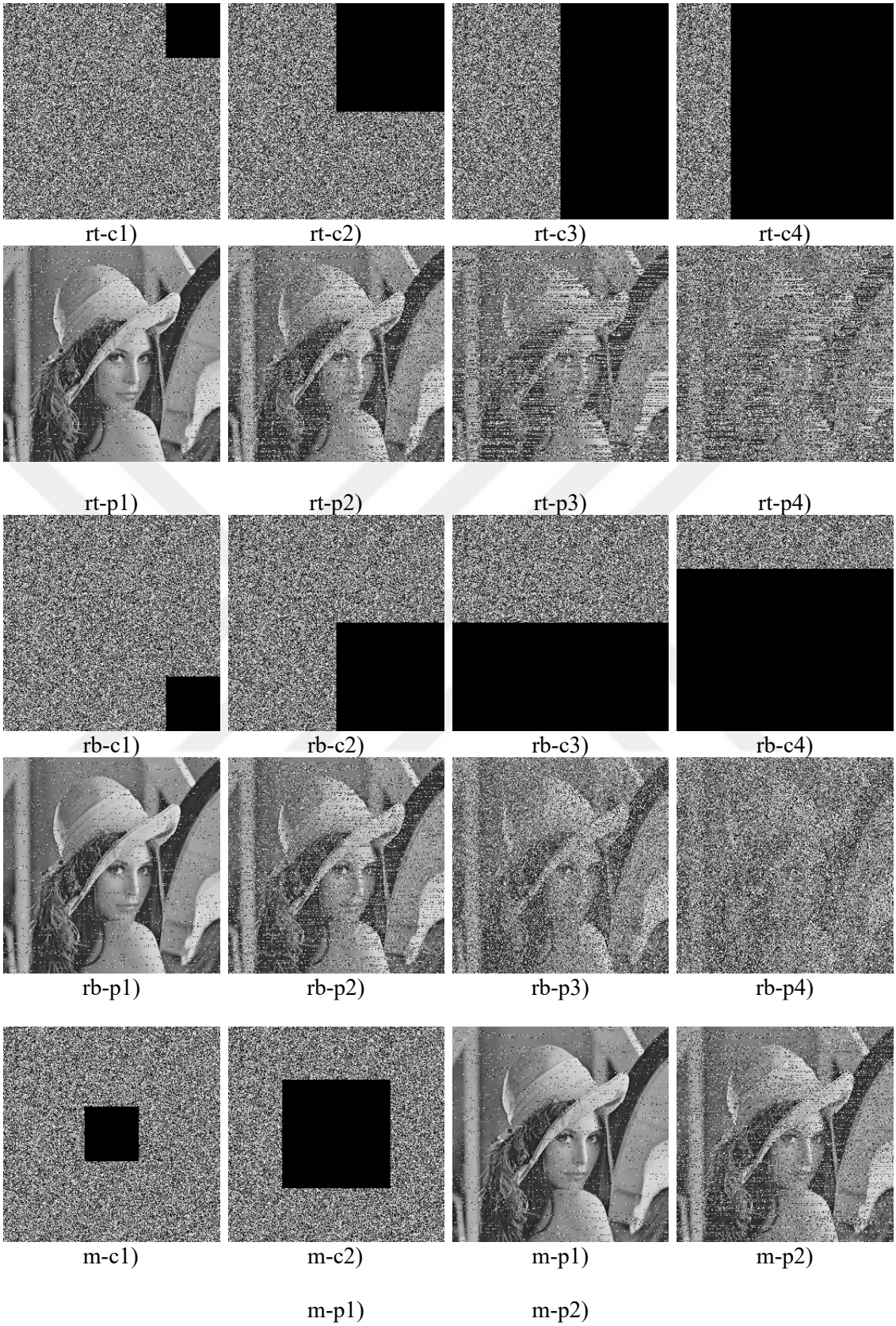
Kırpma saldırılarında şifrelenmiş resim dosyasına ait bazı pikseller tamamen beyaz ya da tamamen siyah renkler ile değiştirilir ve değiştirilmiş bu yeni dosya açılır. Bu işlem şifrelenmiş resim dosyasının farklı bölgelerine farklı büyüklerde yapılır. Başarılı bir şifreleme

algoritmasından bu koşullarda şu iki görevi yerine getirmesi beklenir; a) orijinal resim dosyasına veri kaybına rağmen ulaşılabilmelidir, b) kırpma saldırısı sonrası veri kaybının orijinal dosyanın tamamına dengeli bir şekilde dağılmalıdır.

Bu test kapsamında gri tonlamalı *256x256* boyutlarında "Lena" resim dosyası kullanılmıştır. Öncelikle orijinal dosya belirli bir anahtar kullanılarak şifrelenmiştir. Daha sonra şifrelenmiş resim dosyasının farklı bölgelerine ait farklı sayıdaki piksellerin renkleri tamamen siyah renkler ile değiştirilmiştir. Bölge seçimleri ve piksel sayıları değiştirilerek algoritmanın kırpma saldırılarına karşı dirençliliği zorlanmıştır. Şekil 4.7'de kırpma saldırısı test sonuçları gösterilmiştir. 5 farklı saldırıda resim dosyasının farklı bölgelerine farklı yoğunlukta saldırılar yapılmıştır.



Şekil 4.7: Kırpma saldırısı analizi.



Şekil 4.7 (devam): Kırpma saldırısı analizi.

lt-c1) şifreli resim sol-üst 6.25% kırpma, lt-c2) şifreli resim sol-üst 25% kırpma, lt-c3) şifreli resim sol-üst 50% kırpma, lt-c4) şifreli resim sol-üst 75% kırpma, lt-p1) şifreli resim sol-üst

6.25% kırpma sonrası açma, lt-p2) şifreli resim sol-üst 25% kırpma sonrası açma, lt-p3) şifreli resim sol-üst 50% kırpma sonrası açma, lt-p4) şifreli resim sol-üst 75% kırpma sonrası açma, lb-c1) şifreli resim sol-alt 6.25% kırpma, lb-c2) şifreli resim sol-alt 25% kırpma, lb-c3) şifreli resim sol-alt 50% kırpma, lb-c4) şifreli resim sol-alt 75% kırpma, lb-p1) şifreli resim sol-alt 6.25% kırpma sonrası açma, lb-p2) şifreli resim sol-alt 25% kırpma sonrası açma, lb-p3) şifreli resim sol-alt 50% kırpma sonrası açma, lb-p4) şifreli resim sol-alt 75% kırpma sonrası açma, rt-c1) şifreli resim sağ-üst 6.25% kırpma, rt-c2) şifreli resim sağ-üst 25% kırpma, rt-c3) şifreli resim sağ-üst 50% kırpma, rt-c4) şifreli resim sağ-üst 75% kırpma, rt-p1) şifreli resim sağ-üst 6.25% kırpma sonrası açma, rt-p2) şifreli resim sağ-üst 25% kırpma sonrası açma, rt-p3) şifreli resim sağ-üst 50% kırpma sonrası açma, rt-p4) şifreli resim sağ-üst 75% kırpma sonrası açma, rb-c1) şifreli resim sağ-alt 6.25% kırpma, rb-c2) şifreli resim sağ-alt 25% kırpma, rb-c3) şifreli resim sağ-alt 50% kırpma, rb-c4) şifreli resim sağ-alt 75% kırpma, rb-p1) şifreli resim sağ-alt 6.25% kırpma sonrası açma, rb-p2) şifreli resim sağ-alt 25% kırpma sonrası açma, rb-p3) şifreli resim sağ-alt 50% kırpma sonrası açma, rb-p4) şifreli resim sağ-alt 75% kırpma sonrası açma, m-c1) şifreli resim orta 6.25% kırpma, m-c2) şifreli resim orta 25% kırpma, m-p1) şifreli resim orta 6.25% kırpma sonrası açma, m-p2) şifreli resim orta 25% kırpma sonrası açma.

Şekil 4.7 detaylı bir şekilde incelendiğinde geliştirilmiş olan şifreleme algoritmasının başarılı sonuçlar verdiğini görebiliriz. Kırpma saldırılarına karşı dirençli olmanın iki koşulu olan kırpılmış dosyadan orijinal dosyaya ulaşma ve kırılmış bölgenin orijinal dosya içerisine dengeli bir şekilde dağılmış olma durumları verilen grafiklerden rahatlıkla gözlemlenebilmektedir.

Ayrıca algoritmanın dayanıklılığını zorlamak için şifrelenmiş resim dosyasındaki verinin *75%*'ini kaybederek çeşitli testler yapılmıştır. Normal şartlarda bu büyüklükte bir veri kaybına uğramış dosyadan orijinal veriyi elde etmek bir şifreleme algoritmasından beklenmemektedir. Bu şartlar altında dahi geliştirmiş olduğumuz algoritma orijinal dosyaya ulaşabilmiştir ki bu oldukça başarılı bir sonuçtur.

Tablo 4.4'de çeşitli kırpma saldırıları sonrasında elde edilen PSNR değerleri listelenmiştir. Tablo dikkatli incelendiğinde PSNR değerindeki artış orijinal dosyaya olan benzerliğin arttığı anlamına gelmektedir.

Tablo 4.4: Kırpma saldırısı PSNR değerleri.

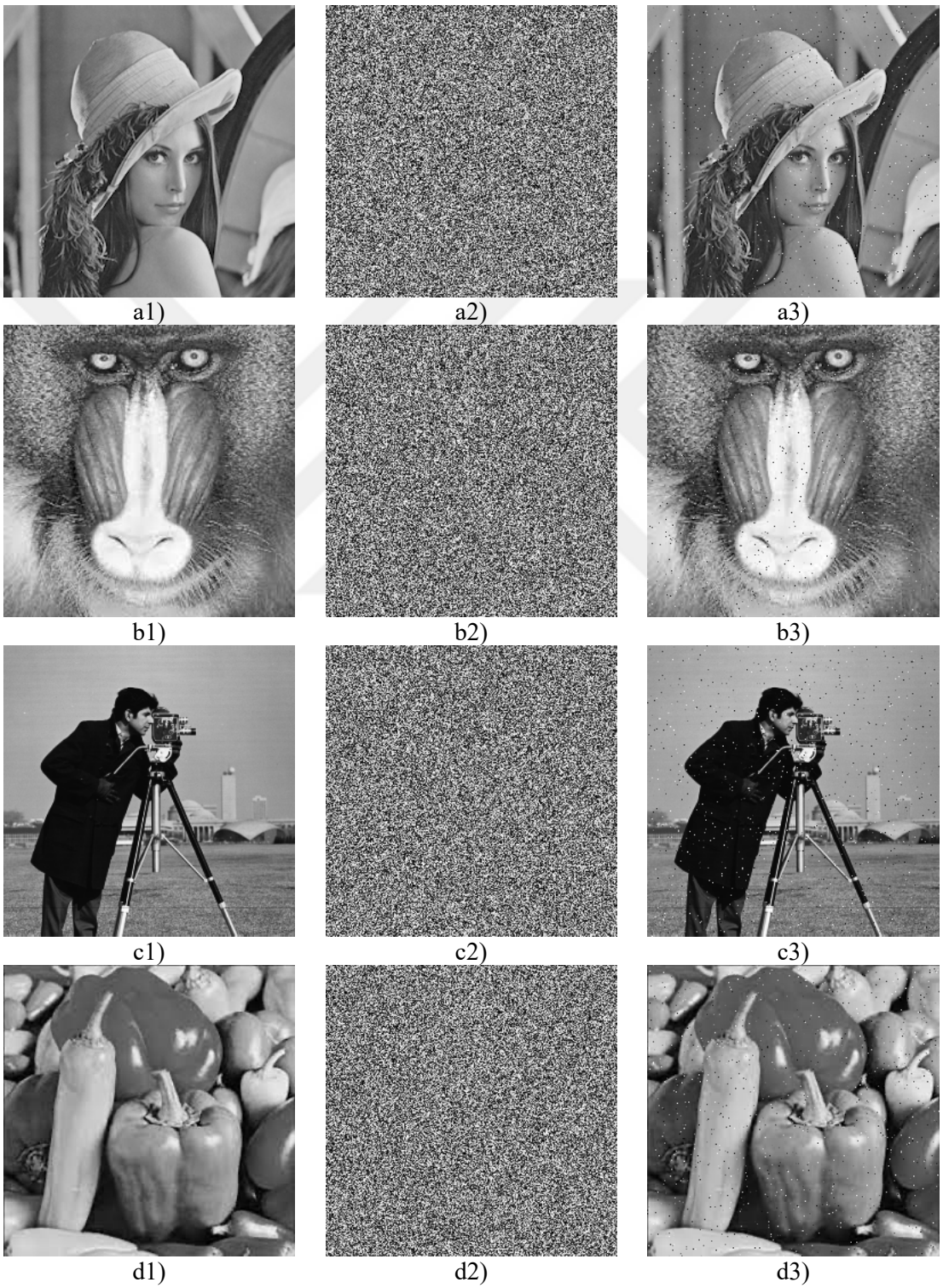
Kırpma Oranı (%)	Bölge	Boyut	PSNR(%)
6.25%	Sol Üst	256x256	39.871294
25%	Sol Üst	256x256	33.908085
50%	Sol Üst	256x256	30.914040
75%	Sol Üst	256x256	29.156877
6.25%	Sol Alt	256x256	39.822301
25%	Sol Alt	256x256	33.856872
50%	Sol Alt	256x256	30.892198
75%	Sol Alt	256x256	29.155073
6.25%	Sağ Üst	256x256	39.935199
25%	Sağ Üst	256x256	33.940656
50%	Sağ Üst	256x256	30.938191
75%	Sağ Üst	256x256	29.172126
6.25%	Sağ Alt	256x256	39.814726
25%	Sağ Alt	256x256	33.904959
50%	Sağ Alt	256x256	30.910539
75%	Sağ Alt	256x256	29.159023
6.25%	Orta	256x256	39.885951
25%	Orta	256x256	33.914039

4.5.2. Gürültü Saldırısı (Noise Attack)

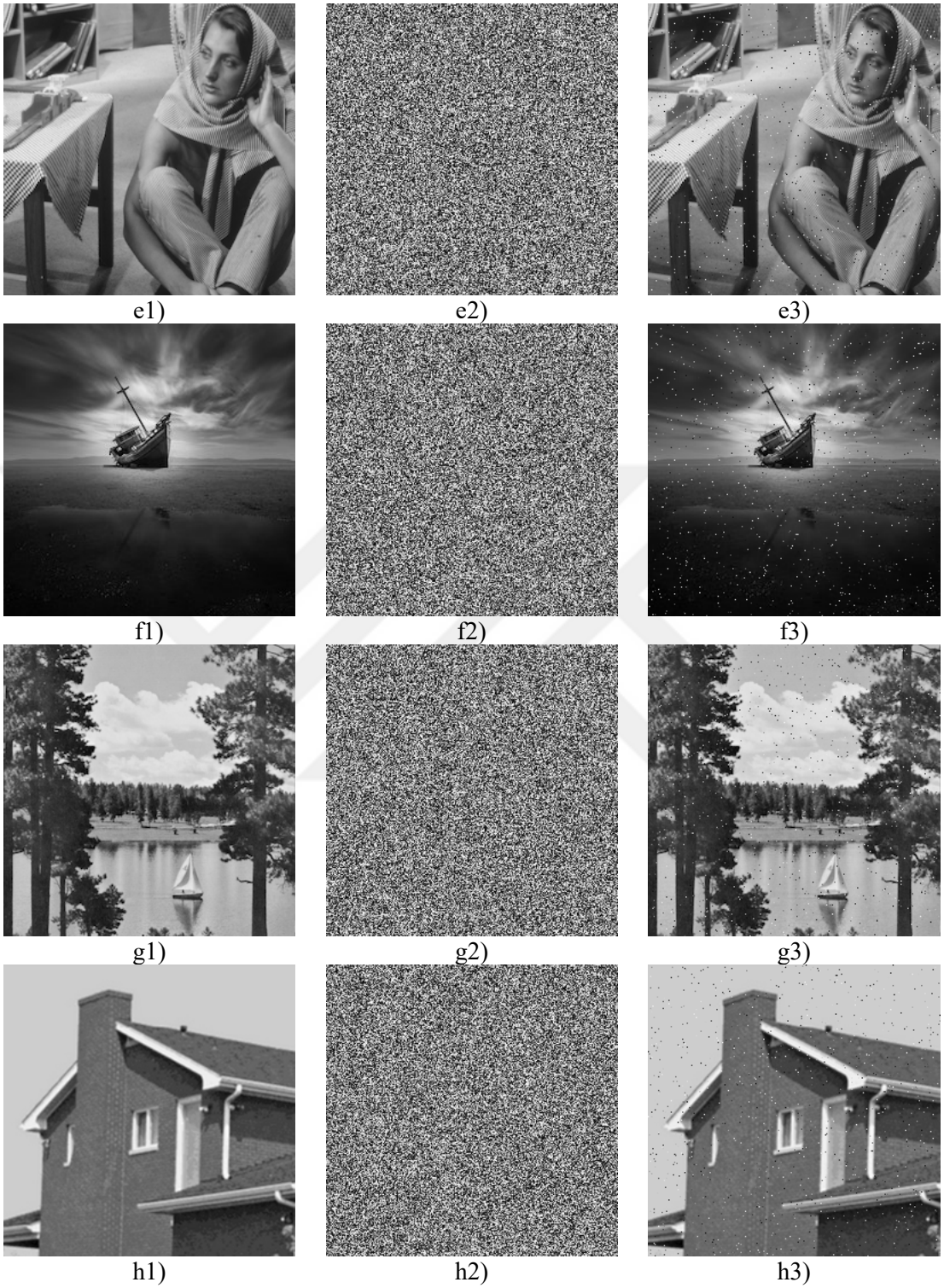
Dayanıklılık analizinde kullanılan bir diğer yöntem ise Gürültü Saldırısıdır (Noise Attack). Bu saldırı şeklinde orijinal dosya öncelikle şifrelenir, şifrelenmiş dosya üzerinde rastgele seçilen piksel değerlerinin renkleri siyah ya da beyaz olacak şekilde değiştirilir. Bu şekilde değiştirilmiş olan yeni şifrelenmiş dosya aynı algoritma ile açılır. Başarılı bir şifreleme algoritmasından bu koşullarda şu iki görevi yerine getirmesi beklenir; a) veri kaybına rağmen orijinal dosyanın elde edilebilmesi, b) bozulmuş piksellerin resim dosyası içerisinde dengeli bir şekilde dağılmış olması.

Gürültü saldırısı analizi kapsamında şu adımlar uygulanmıştır; orijinal resim dosyası şifrelenmiştir. Şifrelenmiş resim dosyasından 1000 ila 1100 arasında rastgele pikseller seçilmiş ve bu piksellerin değerleri siyah olarak değiştirilerek dosya kaydedilmiştir. Bu şekilde

güncellenen yeni dosya şifre açma algoritması ile açılmış ve elde edilen dosya orijinal dosya ile karşılaştırılmıştır. Şekil 4.8'de 8 ayrı resim dosyası için gürültü saldırı test sonuçları verilmiştir. Sonuçlar detaylı incelendiğinde geliştirilen algoritmanın gürültü saldırılarına karşı oldukça dirençli olduğunu söylemek mümkündür.



Şekil 4.8: Gürültü saldırı analizi.



Şekil 4.8 (devam): Gürültü saldırı analizi.

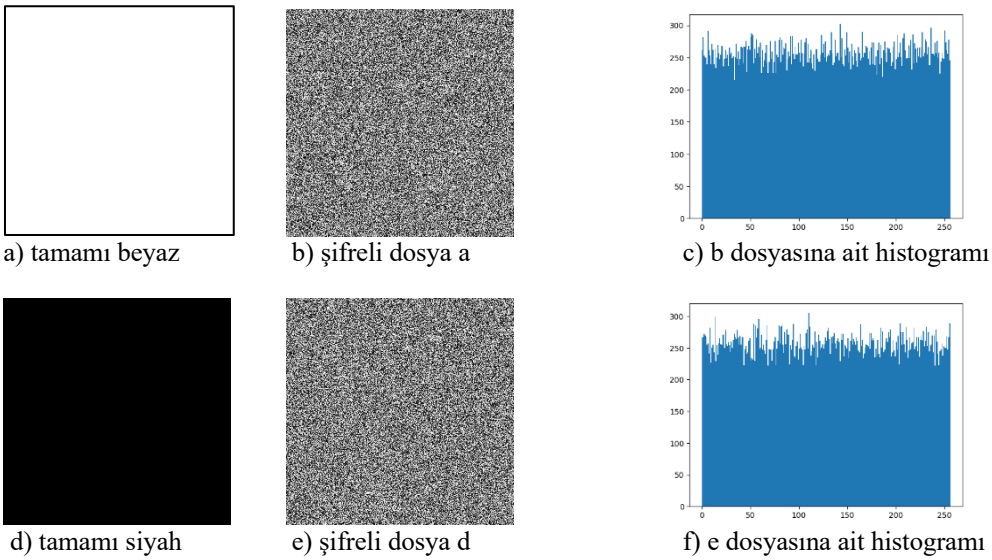
a1) “Lena” orijinal dosyası a2) “Lena” şifrelenmiş dosyaya gürültü eklenmesi, a3) “Lena” bozulmuş dosyanın açılması, b1) “Baboon” orijinal dosyası b2) “Baboon” şifrelenmiş dosyaya gürültü eklenmesi, b3) “Baboon” bozulmuş dosyanın açılması, c1) “Cameraman” orijinal dosyası c2) “Cameraman” şifrelenmiş dosyaya gürültü eklenmesi, c3) “Cameraman” bozulmuş dosyanın açılması, d1) “Peppers” orijinal dosyası d2) “Peppers” şifrelenmiş dosyaya gürültü

eklenmesi, d3) “Peppers” bozulmuş dosyanın açılması, e1) “Barbara” orijinal dosyası e2) “Barbara” şifrelenmiş dosyaya gürültü eklenmesi, e3) “Barbara” bozulmuş dosyanın açılması, f1) “Boat” orijinal dosyası f2) “Boat” şifrelenmiş dosyaya gürültü eklenmesi, f3) “Boat” bozulmuş dosyanın açılması, g1) “Lake” orijinal dosyası g2) “Lake” şifrelenmiş dosyaya gürültü eklenmesi, g3) “Lake” bozulmuş dosyanın açılması, h1) “House” orijinal dosyası h2) “House” şifrelenmiş dosyaya gürültü eklenmesi, h3) “House” bozulmuş dosyanın açılması.

4.5.3. Bilinen Açık Veri ve Seçilen Açık Veri Saldırısı (Known-plaintext and Chosen-plaintext attacks)

Kriptanalistlerin şifrelenmiş veriden orijinal veriye yetkisiz (anahtarsız) ulaşmak için kullandıkları bir saldırı şekli de bilinen açık veri ve seçilen açık veri saldırılarıdır. Bu saldırı şeklinde tamamen beyaz ve tamamen siyah renklerden oluşan resim dosyaları kullanılır. Şifreleme algoritması ile bu tür resim dosyaları şifrelenerek elde edilen şifreli dosyalar üzerinden algoritmanın yapısı, kullanılan teknikler ve gizli anahtar hakkından bilgi elde edilmesi amaçlanmaktadır. Bu kapsamda tamamen siyah ve tamamen beyaz renklerden oluşan *256x256* boyutlarında gri tonlamalı iki ayrı resim dosyası kullanılarak testler yapılmıştır.

Şekil 4.9’da bu kapsamda yapılan analiz sonuçları paylaşılmıştır. Sonuçlar dikkatli incelendiğinde kriptanalistlerin şifrelenmiş dosya üzerinden orijinal veri, anahtar ve algoritma yapısı hakkında bilgi edinmelerinin mümkün olmağı görülmektedir.



Şekil 4.9: Bilinen açık veri ve seçilen açık veri saldırı analizi.

a) Tamamen beyaz renklerden oluşan resim dosyası, b) Tamamen beyaz renklerden oluşan resim dosyasının şifreli hali, c) Tamamen siyah renklerden oluşan resim dosyası, b) Tamamen siyah renklerden oluşan resim dosyasının şifreli hali.

Tablo 4.5’de tamamı beyaz ve tamamı siyah resim dosyaları için korelasyon katsayıları (yatay, dikey, çapraz), ki-kare değerleri, genel ve bölgesel entropi değerleri verilmiştir. Bu

değerler detaylı incelendiğinde önerilen algoritmanın literatürde belirtilen eşik değerlerin üzerinde bir güvenlik seviyesi sunduğu gözlemlenecektir.

Tablo 4.5: Tamamı beyaz ve tamamı siyah dosyalar için güvenlik analizi.

Resim	Korelasyon katsayıları			Ki-kare değeri	Genel entropi	Bölgesel entropi
	Yatay	Çapraz	Dikey			
Tamamı beyaz	0.0015	0.0003	-0.0011	253.81	7.9970	7.9013
Tamamı siyah	-0.0012	0.0014	0.0012	262.15	7.9972	7.9008

5. TARTIŞMA

Hafif sıklet şifreleme algoritmaları son zamanlar araştırmacılar arasında gidererek popülarite kazanan bir çalışma alanıdır. Araştırmacılar geliştirdikleri algoritmalarda sürekli farklı yöntemler deneyerek kaynak kısıtlı sistemlerde çalışabilecek efektif algoritmalar geliştirme çabası içerisindedirler. Kaotik sistemler, sunmuş oldukları özellikler nedeni ile son zamanlarda hafif sıklet şifreleme algoritmalarında yaygın bir şekilde kullanılmaya başlamıştır. Mevcut çalışmalar incelendiğinde bu alanda araştırmacıların yapmış olduğu birçok çalışmayı görmekteyiz. Bu konularda hazırlanan araştırma makalelerinin sayısı her geçen gün artmaktadır. Bütün bu çalışmalara rağmen halen bu alan sıcak bir araştırma sahası olup önümüzdeki yıllarda da bu alanda birçok yeni çalışmanın yapılacağını tahmin ediyoruz.

Bu bölümde, tez kapsamında geliştirmiş olduğumuz kaos tabanlı hafif sıklet şifreleme algoritmasının benzer çalışmalar ile karşılaştırılması yapılmıştır. Bu karşılaştırmalar iki ayrı başlık altında incelenmiştir. Bunlar; kaos tabanlı diğer şifreleme algoritmaları ile olan güvenlik ve performans karşılaştırmaları ve diğer hafif sıklet şifreleme algoritmaları ile olan zaman karmaşıklığı karşılaştırmalarıdır.

5.1. GÜVENLİK VE PERFORMANS KARŞILAŞTIRMALARI

Çavuşoğlu ve diğ. [98], yapmış oldukları çalışmada kaos tabanlı S-BOX kullanarak yeni bir algoritma geliştirmişlerdir. Bu geliştirmiş oldukları algoritmanın performansını ölçmek için benzer algoritmalar ile karşılaştırmalar yapmışlar. Bunun için *256x256* gri tonlamalı "Crowd" resim dosyası kullanılmıştır. Geliştirmiş olduğumuz algoritmayı aynı resim dosyasını kullanarak Çavuşoğlu ve diğ. [98], elde etmiş olduğu sonuçlar ile karşılaştırdık. Sonuçlar Tablo 5.1'de sunulmuştur.

Tablo 5.1: Şifreleme algoritmaları güvenlik ve performans karşılaştırmaları 1.

Algoritma	**NPCR (%)**	**UACI (%)**	**Bilgi Entropisi**
Kaotik Algoritma [98]	99.60672	31.24768	7.95454
AES Algoritması [98]	99.63245	31.87236	7.95912
Ref. Algoritma [98]	99.62987	31.83459	7.95667
Önerilen Algoritma	99.60242	33.54216	7.99690

Sonuçlar detaylı bir şekilde incelendiğinde NPCR değeri bakımında geliştirmiş olduğumuz algoritma diğer algoritmalara yakın sonuçlar vermişken UACI ve Bilgi Entropisi kriterlileri bakımdan diğer şifreleme algoritmalarından çok daha iyi sonuçlar vermiştir. Diğer bir karşılaştırmamız da korelasyon katsayıları, NPCR, UACI ve Bilgi Entropisi parametreleri üzerinden yapılmıştır. Diğer algoritmalar ile uygun bir karşılaştırma yapabilmek için *512x512* gri tonlamalı "Lena" resim dosyası kullanılmıştır. Sonuçlar Tablo 5.2'de gösterildiği gibidir.

Tablo 5.2: Şifreleme algoritmaları güvenlik ve performans karşılaştırmaları 2.

Algoritma	**Korelasyon Katsayısı**			**Bilgi Entropisi**	**NPCR(%)**	**UACI(%)**
	Yatay	**Çapraz**	**Dikey**			
Ref. [99]	-0.0013	0.0018	0.0023	7.9974	0.9960	0.3339
Ref. [100]	-0.0029	0.0045	0.0015	7.9970	0.9961	0.3338
Ref. [56]	0.0692	0.0544	0.0396	7.8152	0.9642	0.2735
Ref. [50]	-0.0063	0.0095	0.0089	7.9974	0.9946	0.3764
Ref. [101]	-0.0045	0.0053	-0.0296	7.9993	0.9959	0.3341
Ref. [52]	-0.0245	-0.0193	-0.0226	7.9972	0.9960	0.2862
Ref. [102]	0.0033	-0.0002	-0.0008	-	0.9954	0.3377
Önerilen Algoritma	0.0009	-0.0095	0.0078	7.9992	0.9960	0.3343

Tablo detaylı bir şekilde incelendiğinde, korelasyon katsayıları açısından bakıldığında sunmuş olduğumuz şifreleme algoritmasının ulaştığı değerler tabloda sunulan diğer çalışmaların korelasyon katsayısı değerlerinin ortalamasından daha iyidir. Bilgi entropisi açısından bakıldığında geliştirilen algoritma en iyi ikinci değeri elde etmiştir. NPCR ve UACI değerleri açısından bakıldığında geliştirilen algoritma karşılaştırmaya konu olan diğer algoritmaların ortalama değerlerinin üzerinde değerlere ulaşmıştır. Karşılaştırmaya konu olan

bütün parametreler açısında bakıldığında önerilen algoritma ortalamanın çok üzerinde bir başarı elde etmiştir.

5.2. ZAMAN KARMAŞIKLIĞI KARŞILAŞTIRMALARI

Önerilen hafif sıklet şifreleme algoritması zaman karmaşıklığı açısından yaygın olarak bilinen ve kullanılan diğer hafif sıklet şifreleme algoritmaları ile karşılaştırılmıştır. Bu kapsamda Tea, xTea, Simon, Speck, Katan, Lea, Present, Aes, Des hafif sıklet şifreleme algoritmaları önerilen algoritma ile karşılaştırılmıştır. Her bir algoritmanın python kaynak kodu referans olarak verilen internet sayfalarından alınıp test makinasında derlenmiş, karşılaşılan bazı ufak sorunlar çözülmüş ve çalışır hala getirilmiştir.

Karşılaştırmalar *512x512* gri tonlamalı "Lena" resim dosyası üzerinden yapılmıştır. Her bir algoritmanın python uyarlaması i7 işlemcili, 8 GiB hafızaya sahip Mac OS X platformunda çalıştırılmıştır. Sonuçlar Tablo 5.3'de listelendiği gibidir.

Tablo 5.3: Şifreleme algoritmaları zaman karmaşıklık analizi.

Algoritma	**Şifreleme Süresi (ms)**	**Şifre Çözme Süresi (ms)**	**Toplam Süre (ms)**
Tea [103]	581.3560	621.6370	1202.9929
xTea [104]	1774.6841	1563.5559	3338.2400
Simon [105]	1515.4307	1476.8621	2992.2927
Speck [105]	803.4417	935.0908	1738.5325
Katan [106]	81962.7583	84619.9905	166582.7488
Lea [107]	24072.1958	23775.7605	47847.9563
Present [108]	20583.6730	21682.1703	42265.8433
AES [109]	3553.9932	3575.2778	7129.2710
DES [110]	25422.1689	25417.6970	50839.8660
Önerilen Algoritma	739.9189	899.5991	1639.5181

Karşılaştırma tablosu detaylı incelendiğinde geliştirmiş olduğumuz hafif sıklet şifreleme algoritmasının Tea algoritmasından sonra en hızlı şifreleme ve şifre çözme algoritması olduğu görülmektedir. DES algoritmasından yaklaşık olarak 31 kat daha hızlı, AES algoritmasından yaklaşık olarak 4.5 kat daha hızlıdır. Tea algoritmasında yaşanan zafiyetler nedeniyle xTea (eXtended TEA - Genişletilmiş TEA) algoritması geliştirilmiş. Bu bağlamda

değerlendirdiğimiz zaman önerilen algoritma kaynak kısıtlı sistemler için oldukça güçlü bir güvenlik algoritması alternatifi olarak düşünülebilir. Tablo 5.4'de geliştirilen algoritmanın Büyük O notasyonuna (Big O notation) göre zaman karmaşıklığı modüller bazında kırılımlı olarak gösterilmiştir. Buna göre geliştirilen algoritmanın toplam zaman karmaşıklığı $2N^2 + 4N$'dir.

Tablo 5.4: Geliştirilen algoritma zaman karmaşıklığı.

Modül	**Zaman Karmaşıklığı**
calculateHashValue	N^2
confusion	2N
diffusion	N^2
Hide & Swap H2 operation	2N
Total	$2N^2 + 4N$

Tablo 5.5'de ise geliştirilen algoritmanın benzer çalışmalar ile zaman karmaşıklığı bakımında Büyük O notasyonuna göre karşılaştırılması verilmiştir. Tablo detaylı bir şekilde incelendiğinde tez kapsamında sunulan algoritmanın $2N^2 + 4N$ zaman karmaşıklık değeri ile benzer çalışmalardan çok daha iyi bir değer sunduğu görülecektir.

Tablo 5.5: Algoritmaların zaman karmaşıklığı karşılaştırması.

Algoritma	**Zaman Karmaşıklığı**
Ref. [111]	$3N^2$
Ref. [112]	$6N^2$
Ref. [113]	$5N^2 + N$
Ref. [114]	$5.5N^2$
Ref. [115]	$4(N + 2)^2 + 4(N + 2)^2 \log (N + 2)$
Önerilen Algoritma	$2N^2 + 4N$

Bütün test sonuçlarını toplu olarak değerlendirdiğimizde tez kapsamında sunmuş olduğumuz hafif sıklet şifreleme algoritması gerek güvenlik ve performans gerekse hız bakımından kaynak kısıtlı sistemler için oldukça iddialı bir alternatif olarak karşımıza çıkmaktadır. Algoritmamız birçok karşılaştırma parametresi bakımında en iyi sonuçları veren ilk üç algoritma arasına girmeyi başarmıştır.

6. SONUÇ VE ÖNERİLER

Teknolojinin hızla gelişmesi ve dijital verinin hayatımızın her alanına girmesiyle birlikte veri güvenliği daha önemli bir konu olarak karşımıza çıkmaktadır. Özellikli IoT, giyilebilir cihazlar, kablosuz sensor ağlar, RFID etiketler gibi sınırlı kaynaklara sahip sistemler için AES, DES, 3DES, RSA gibi geleneksel çözümler kaynak gereksinimi açısından uygun düşmemektedirler. Bu durum daha az kaynak tüketen ve üst düzeyde güvenlik sunan hafif sıklet şifreleme algoritmalara olan ihtiyacı daha fazla gündeme getirmektedir.

Önerilen kaos tabanlı hafif sıklet şifreleme algoritması kaynak kısıtlı sistemler için bir güvenlik çözümüdür. Simetrik şifreleme algoritması yapısında olan algoritmada 2 farklı lojistik harita kullanılmıştır. İlk lojistik haritanın başlangıç parametreleri doğrudan gizli anahtardan türetilmiş olup, sunmuş olduğu anahtar dizileri karıştırma, dağıtma ve maskeleme işlemlerinde kullanılmıştır. Bu sayede algoritmanın özellikle kırpma saldırılarına karşı dayanıklılığı artmıştır. Ayrıca pikseller arasındaki yatay, dikey, çapraz bütün korelasyonlar bozularak istatistiksel saldırıların önüne geçilmiştir. Testlerde elde edilen orijinal dosya ve şifrelenmiş dosya korelasyon katsayıları bu gerçeği doğrulamaktadır. İkinci haritanın başlangıç parametreleri gizli anahtar ve orijinal dosyanın özet kodundan türetilmiştir ve türetilen anahtar dizisi XoR işleminde (şifreleme) kullanılmıştır. Bu sayede algoritma diferansiyel saldırılara karşı dayanıklılık kazanmıştır. Analiz ve testlerde elde edilen NPCR ve UACI değerleri bunu doğrulamaktadır.

İkinci kaotik haritanın başlangıç parametrelerinin belirlenmesinde kullanılan orijinal dosyaya ait özet koda şifre çözme aşamasında ihtiyaç duyulacaktır. Bu amaçla özet kod değerinin şifrelenmiş dosya içerisinde gizlenmesi gerekmektedir. Özet kod olmadan orijinal dosyaya dönüş sağlanamayacağı için özet kodun güvenli bir şekilde dosya içerisinden elde edilebilmesi elzemdir. Bunun için özet koda bir doğrulama değeri eklenmiştir. Bu doğrulama değeri özet kodun özellikle kırpma ve gürültü saldırılarından etkilenmemesi için güvenlik amacıyla eklenmiştir. Elde edilen yeni veri bloğu şifrelemiş resim dosyası içerisine birinci kaotik haritadan elde edilen rastgele veriler kullanılarak resim dosyası kolon uzunluğunun üçte biri sayıda farklı konumlara dağıtılmıştır. Yine veri güvenliğinin arttırılması için bu veri bloğu birinci kaotik haritadan elde edilen veriler ile XoR işleminden geçirilmiştir. Bu veri bloğuna

yer açabilmek için dosyanın sonuna yeni bir satır eklenmiş ve değiş tokuş işlemlerinde elde edilen şifrelenmiş piksel değerleri bu yeni satıra sırası ile yazılmıştır.

Geliştirilen şifreleme algoritması güvenli bir şifreleme algoritmasının sağlaması gereken bütün kriterler bakımından testlere tabi tutulmuştur. Testler farklı farlı birçok resim dosyası ile tekrar tekrar yapılmıştır. Elde edilen değerler her bir kriter için belirlenen asgari değer ile karşılaştırılmış ve herhangi bir zafiyet olmadığı teyit edilmiştir.

Asgari 100 bit olması gereken anahtar uzayı analizini 196 bit ile geçmiştir. Anahtar hassasiyet analizini ilk anahtarda yapılan 10^{-15}'lik ufak bir değişiklik ile başarılı bir şekilde geçmiştir. 256 farklı renk sıkalasına sahip resim dosyaları için minimum 7.99 olarak belirtilen bilgi entropisi analizini, ortalama 7.997 ile 8 olan ideal değere oldukça yakın bir ortalamayla başarılı bir şekilde geçmiştir. Histogram analizinde dalgalı olan orijinal resim dosyası histogramı şifrelenmiş resim dosyaları için beklendiği gibi yatay bir değere dönüşmüştür. Korelasyon analizinde teste tabi tutulan bütün dosyalar için yatay, dikey ve çapraz korelasyonlara tek tek bakılmıştır. Korelasyon katsayıları her üç korelasyon tipi için 10000 piksel ile detaylı bir şekilde incelenmiştir. Elde edilen korelasyon katsayıları benzer çalışmalar ile karşılaştırılmış ve ortalamanın üzerinde değerler elde edildiği gözlemlenmiştir.

Orijinal resim dosyasında bir bit değiştirilerek elde edilen şifrelenmiş dosyaların karşılaştırmasına dayanan diferansiyel saldırılar bakımından bütün test resim dosyaları incelenmiştir. Her bir dosya için NPCR ve UACI değerleri hesaplanmıştır. Başarılı bir şifreleme algoritması için belirtilen asgari NPCR değeri 99%'dır, UACI değerin ise 33%'e yakınsamalıdır. Ortamla 99.6% NPCR değeri ve yine ortalama 33.4% UACI değerleri elde edilmiştir ki bu değerler benzer çalışmalara göre oldukça başarılı sonuçlardır.

Kırpma saldırısı açısında yapılan testlerde şifrelenmiş resim dosyasının farklı bölgelerinden (sol-üst, sol-alt, sağ-üst, sağ-alt, orta) farklı oranlarda (orijinal dosyanın 6.25%, 25%, 50%, 75% oranlarında (mevcut çalışmalarda 75% oranında bir kırpma testi yapıldığı gözlemlenmemiştir)) kırpmalar yapılmıştır. Kırpmaya uğramış resim dosyaları başarılı bir şekilde açılmıştır. Kırpma yapılan bölgenin beklendiği gibi açılan dosyanın bütüne yayıldığı gözlemlenmiştir. Aynı şekilde gürültü saldırıları [1000-1100] aralığındaki piksellere rastgele olarak uygulanmıştır ve gürültü saldırısına uğramış dosyalar başarılı bir şekilde açılabilmiştir. Son adımda bilinen ve seçilen düz metin saldırı testleri tamamı siyah renklerden oluşan ve tamamı beyaz renklerden oluşan dosyalar ile yapılmış ve algoritmanın dayanaklılığı başarılı bir

şekilde gözlemlenmiştir. Bütün bu aşamalarda elde edilen veriler benzer çalışmalar ile detaylı bir şekilde karşılaştırılmıştır. Geliştirmiş olduğumuz algoritma hemen hemen bütün kriterler bakımında ilk üç sırada kendine yer bulmuştur.

Son aşamada, önerilen algoritma benzer hafif sıklet şifreleme algoritmaları (Tea, xTea, Simon, Speck, Katan, Lea, Present, Aes, Des) ile zaman karmaşıklığı analizine tabi tutulmuştur. Bu amaçla her bir algoritmanın python dilinde yazılmış kaynak kodları elde edilmiş ve test makinasında çalıştırılmıştır. Her bir algoritmanın şifreleme süresi, şifre çözme süresi ve toplam süreleri hesaplanmıştır. Önerilen algoritma, zafiyetleri ile bilinen Tea algoritmasından sonra en hızlı algoritma olmuştur. Ayrıca $2N^2 + 4N$ zaman karmaşıklığı benzer çalışmaların üzerinde bir değer elde etmiş ve uygun bir hafif sıklet şifreleme algoritması olduğunu göstermiştir.

Bütün bu test ve analiz çalışmalarının sonunda bu tez çalışması kapsamında geliştirilen kaos tabanlı hafif sıklet şifreleme algoritması kaynak kısıtlı sistemler için oldukça iyi sonuçlar bir alternatif olduğu gözlenmiştir.

Gelecek çalışmalarımız için önerilen algoritmanın farklı matematiksel modeller ile optimizasyonunu yapmayı planlıyoruz. Bu sayede daha başarılı sonuçlar alınabileceğini düşünüyoruz. Ayrıca dosya sonuna yeni satır ekleyerek sağladığımız ekstra veri ekleme işlemini, RDH (Reversible Data Hiding – Geri Dönüşümlü Veri Saklama) teknikleri ile yeni satıra ihtiyaç duymadan şifrelenmiş dosya içerisinde saklamayı amaçlıyoruz. Son olarak tez kapsamında geliştirilen hafif sıklet şifreleme algoritmasını gerçek bir kaynak kısıtlı sistem üzerinde uygulanması ve sonuçların analiz edilmesi planlanmaktadır.

KAYNAKLAR

[1] Naik, R. B. and Singh, U., 2022, A review on applications of chaotic maps in pseudo-random number generators and encryption, Annals of Data Science, 11(1), 25-50.

[2] Alassaf, N. and Gutub, A., 2019, Simulating light-weight-cryptography implementation for IOT Healthcare Data Security Applications, International Journal of E-Health and Medical Communications, 10(4), 1-15.

[3] Sleem, L.andCouturier, R., 2020, Speck-R: An ultra light-weight cryptographic scheme for internet of things, Multimedia Tools and Applications, 80(11), 17067-17102.

[4] Shankar, K. and Elhoseny, M., 2019, An optimal light weight cryptography—simon block cipher for secure image transmission in wireless sensor networks, Lecture Notes in Electrical Engineering, 17-32.

[5] Zhang, X. and Ye, R., 2020, A novel RGB image encryption algorithm based on DNA sequences and Chaos, Multimedia Tools and Applications, 80(6), 8809-8833.

[6] Wang, X., Wang, Y., Zhu, X. and Unar, S., 2019, Image encryption scheme based on chaos and DNA plane operations, Multimedia Tools and Applications, 78(18), 26111-26128.

[7] Farah, M. B., Guesmi, R., Kachouri, A. and Samet, M., 2020, A novel chaos based optical image encryption using fractional Fourier transform and DNA sequence operation, Optics & Laser Technology, Cilt 121, 105777.

[8] Chai, X., Fu, X., Gan, Z., Lu, Y. and Chen, Y., 2019, A color image cryptosystem based on dynamic DNA encryption and Chaos, Signal Processing, Cilt 155, 44-62.

[9] Malik, M. G., Bashir, Z., Iqbal, N. and Imtiaz, M. A., 2020, Color image encryption algorithm based on hyper-chaos and DNA computing, IEEE Access, Cilt 8, 88093-88107.

[10] Liu, Z., Wu, C., Wang, J. and Hu, Y., 2019, A color image encryption using dynamic DNA and 4-D memristive hyper-chaos, IEEE Access, Cilt 7, 78367-78378.

[11] Liu, Q. and Liu, L., 2020, Color image encryption algorithm based on DNA coding and Double Chaos System, IEEE Access, Cilt 8, 83596-83610.

[12] Mondal, B. and Mandal, T., 2017, A light weight secure image encryption scheme based on Chaos & DNA Computing, Journal of King Saud University - Computer and Information Sciences, 29(4), 499-504.

[13] Fang, P., Liu, H., Wu, C. and Liu, M., 2021, A secure chaotic block image encryption algorithm using generative adversarial networks and DNA sequence coding, Mathematical Problems in Engineering, Cilt 2021, 1-26.

[14] Chai, X., Chen, Y. and Broyde, L., 2017, A novel chaos-based image encryption algorithm using DNA sequence operations, Optics and Lasers in Engineering, Cilt 88, 197-213.

[15] Zhao, J., Wang, S. and Zhang, L., 2023, Block Image Encryption algorithm based on novel chaos and DNA encoding, Information, 14(3), 150.

[16] Al Shebli, H. M. and Beheshti, B. D., 2018, Light weight cryptography for resource constrained IOT devices, 13-14 November 2018 Vancouver, Canada, 196-204.

[17] Kocarev, L. and Lian, S., 2011, ISBN: 978-3-642-20541-5. Chaos-Based Cryptography, Berlin: Springer.

[18] Paar, C. and Pelzl, J., 2010, ISBN: 978-3-642-04100-6. Understanding Cryptography, Berlin: Springer.

[19] Arafath, M. S., Ur Rahman Khan, K. and Sunitha, K., 2020, Security in Opportunistic Sensor Network and IOT having sensors using light weight key generation and cryptographic algorithm, 27-28 July 2018 Bhubaneswar, India.

[20] Mokhtar, M. A., Gobran, S. N. and El-Badawy, E.-S. A.-M., 2015, Colored image encryption algorithm using DNA code and Chaos Theory, 23-25 September 2014 Kuala Lumpur, Malaysia.

[21] Zhou, S., Wei, Y., Zhang, Y. and Teng, L., 2023, Novel chaotic image cryptosystem using dynamic DNA coding, International Journal of Modern Physics C, Cilt 35(06).

[22] Yildirim, M., 2020, DNA encoding for RGB image encryption with memristor based neuron model and chaos phenomenon, Microelectronics Journal, Cilt 104, 104878.

[23] Zhang, X., Wang, L., Zhou, Z. and Niu, Y., 2019, A chaos-based image encryption technique utilizing Hilbert curves and H-fractals, IEEE Access, Cilt 7, 74734-74746.

[24] Wang, S., Wang, C. and Xu, C., 2020, An image encryption algorithm based on a hidden attractor chaos system and the knuth–durstenfeld algorithm, Optics and Lasers in Engineering, Cilt 128, 105995.

[25] Silva Garcia, V. M., Gonzalez Ramirez, M. D., Carapia, R. F., Vega-Alvarado, E. and Rodriguez Escobar, E., 2019, A novel method for image encryption based on chaos and transcendental numbers, IEEE Access, Cilt 7, 163729-163739.

[26] Alibrahim, H. and Ludwig, S. A., 2022, Image encryption algorithm based on particle swarm optimization and chaos logistic map, 05-07 December 2021 Orlando, FL, USA.

[27] Machkour, M., Saaidi, A. and Benmaati, M. L., 2015, A novel image encryption algorithm based on the two-dimensional logistic map and the Latin Square Image cipher, 3D Research, 6(4).

[28] Rehman, A. U., Firdous, A., Iqbal, S., Abbas, Z., Shahid, M. M., Wang, H. and Ullah, F., 2020, A color image encryption algorithm based on one time key, chaos theory, and concept of Rotor Machine, IEEE Access, Cilt 8, 172275-172295.

[29] Zhao, H. and Njilla, L., 2019, Hardware assisted Chaos based IOT authentication, 09-11 May 2019 Banff, AB, Canada.

[30] Jayaram, A. and Deb, S., 2020, A hybrid addition chaining based light weight security mechanism for enhancing quality of service in IOT, Wireless Personal Communications, 113(2), 1073-1095.

[31] Kifouche, A., Azzaz, M. S., Hamouche, R. and Kocik, R., 2022, Design and implementation of a new lightweight chaos-based cryptosystem to secure IOT Communications, International Journal of Information Security, 21(6), 1247-1262.

[32] Gafsi, M., Hajjaji, M. A., Malek, J. and Mtibaa, A., 2021, FPGA hardware acceleration of an improved chaos-based cryptosystem for real-time image encryption and decryption, Journal of Ambient Intelligence and Humanized Computing, 14(6), 7001-7022.

[33] Koyuncu, İ., Tuna, M., Pehlivan, İ., Fidan, C. B. and Alçın, M., 2019, Design, FPGA implementation and statistical analysis of Chaos-ring based dual entropy core true random number generator, Analog Integrated Circuits and Signal Processing, 102(2), 445-456.

[34] Sharafi, M., Fotouhi-Ghazvini, F., Shirali, M. and Ghassemian, M., 2019, A low power cryptography solution based on chaos theory in wireless sensor nodes, IEEE Access, Cilt 7, 8737-8753.

[35] Senouci, M. R., Sadoudi, S., Djamaa, B. and Senouci, M. A., 2019, A lightweight efficient chaos-based cryptosystem for constrained-node networks, International Journal of Communication Systems, 33(10).

[36] Kiran, H. E., Akgul, A. and Yildiz, O., 2022, A new chaos-based lightweight encryption mechanism for microcomputers, 06-07 June 2022 Istanbul, Turkey.

[37] Sadaghiani, S. and Zolfy, M., 2020, Implementing lightweight image/video encryption cores on Xilinx Zynq, 04-06 August 2020 Tabriz, Iran.

[38] Oteko Tresor, L. and Sumbwanyambe, M., 2019, A selective image encryption scheme based on 2D DWT, Henon map and 4D qi hyper-chaos, IEEE Access, Cilt 7, 103463-103472.

[39] Liu, Y., Jiang, Z., Xu, X., Zhang, F. and Xu, J., 2020, Optical image encryption algorithm based on hyper-chaos and public-key cryptography, Optics & Laser Technology, Cilt 127, 106171.

[40] Bhattacharjee, S., Gupta, M. and Chatterjee, B., 2022, Time efficient image encryption-decryption for visible and COVID-19 X-ray images using modified Chaos-based logistic map, Applied Biochemistry and Biotechnology, 195(4), 2395-2413.

[41] Lin, C.-H., Hu, G.-H., Chen, J.-S., Yan, J.-J. and Tang, K.-H., 2022, Novel design of Cryptosystems for video/audio streaming via Dynamic Synchronized Chaos-based Random Keys, Multimedia Systems, 28(5), 1793-1808.

[42] Ahmad, I. and Shin, S., 2021, A novel hybrid image encryption–compression scheme by combining chaos theory and number theory, Signal Processing: Image Communication, Cilt 98, 116418.

[43] Çavuşoğlu, Ü., Kaçar, S., Pehlivan, I. and Zengin, A., 2017, Secure image encryption algorithm design using a novel chaos based S-box, Chaos, Solitons & Fractals, Cilt 95, 92-101.

[44] Wang, X., Liu, C. and Zhang, H., 2016, An effective and fast image encryption algorithm based on chaos and interweaving of ranks, Nonlinear Dynamics, 84(3), 1595-1607.

[45] Faragallah, O. S., 2014, Efficient confusion–diffusion chaotic image cryptosystem using enhanced standard map, Signal, Image and Video Processing, 9(8), 1917-1926.

[46] Bakhshandeh, A. and Eslami, Z., 2013, An authenticated image encryption scheme based on chaotic maps and memory cellular automata, Optics and Lasers in Engineering, 51(6), 665-673.

[47] Hua, Z. and Zhou, Y., 2017, Design of image cipher using block-based scrambling and image filtering, Information Sciences, Cilt 396, 97-113.

[48] Zhang, Y. and Tang, Y., 2017, A plaintext-related image encryption algorithm based on Chaos, Multimedia Tools and Applications, 77(6), 6647-6669.

[49] Bhatt, D. P., Raja, L. and Sharma, S., 2020, Light-weighted cryptographic algorithms for energy efficient applications, Journal of Discrete Mathematical Sciences and Cryptography, 23(2), 643-650.

[50] Kamal, M. and Tariq, M., 2019, Light-weight security for advanced metering infrastructure, 28 April 2019 - 01 May 2019 Kuala Lumpur, Malaysia.

[51] Shi, Y., Wei, W., Fan, H., Au, M. H. and Luo, X., 2019, A light-weight white-box encryption scheme for securing distributed embedded devices, IEEE Transactions on Computers, 68(10), 1411-1427.

[52] Özkaynak, F., 2018, Brief review on application of Nonlinear Dynamics in image encryption, Nonlinear Dynamics, 92(2), 305-313.

[53] Teh, J. S., Alawida, M. and Sii, Y. C., 2020, Implementation and practical problems of Chaos-based cryptography revisited, Journal of Information Security and Applications, Cilt 50, 102421.

[54] Tutueva, A. V., Nepomuceno, E. G., Karimov, A. I., Andreev, V. S. and Butusov, D. N., 2020, Adaptive chaotic maps and their application to pseudo-random numbers generation, Chaos, Solitons & Fractals, Cilt 133, 109615.

[55] Alawida, M., Teh, J. S., Oyinloye, D. P., Alshoura, W. H., Ahmad, M. and Alkhawaldeh, R. S., 2020, A new hash function based on chaotic maps and deterministic finite state automata, IEEE Access, Cilt 8, 113163-113174.

[56] Alcin, M., Koyuncu, I., Tuna, M., Varan, M. and Pehlivan, I., 2018, A novel high speed artificial neural network–based chaotic true random number generator on Field Programmable Gate Array, International Journal of Circuit Theory and Applications, 47(3), 365-378.

[57] Alawida, M., Samsudin, A., Alajarmeh, N., Teh, J. S., Ahmad, M. and Alshoura, W. H., 2021, A novel hash function based on a chaotic sponge and DNA sequence, IEEE Access, Cilt 9, 17882-17897.

[58] Zaher, A. A. and Abu-Rezq, A., 2011, On the design of Chaos-based Secure Communication Systems, Communications in Nonlinear Science and Numerical Simulation, 16(9), 3721-3737.

[59] Masood, F., Driss, M., Boulila, W., Ahmad, J., Rehman, S. U., Jan, S. U., Qayyum, A. and Buchanan, W. J., 2021, A lightweight chaos-based medical image encryption scheme using random shuffling and XOR operations, Wireless Personal Communications, 127(2), 1405-1432.

[60] Fawaz, Z., El Assad, S., Frajallah, M., Khalil, A., Lozi, R. and Deforges, O., 2014, Lightweight Chaos-based cryptosystem for secure images, 09-12 December 2013 London, UK.

[61] Ferdush, J., Begum, M. and Uddin, M. S., 2021, Chaotic lightweight cryptosystem for image encryption, Advances in Multimedia, Cilt 2021, 1-16.

[62] Atawneh, B., AL-Hammoury, L. and Abutaha, M., 2020, Power consumption of a chaos-based stream cipher algorithm, 19-21 March 2020 Riyadh, Saudi Arabia.

[63] Alawida, M., Samsudin, A., Teh, J. S. and Alshoura, W. H., 2019, Digital cosine chaotic map for cryptographic applications, IEEE Access, Cilt 7, 150609-150622.

[64] Wang, X., Xue, W. and An, J., 2020, Image encryption algorithm based on tent-dynamics coupled map lattices and diffusion of household, Chaos, Solitons & Fractals, Cilt 141, 110309.

[65] Zhao, F., Li, C., Liu, C. and Zhang, J., 2019, Image encryption algorithm based on Sine-Logistic Cascade Chaos, 19-22 April 2019 Beijing, China.

[66] Zhang, Y., 2020, The fast image encryption algorithm based on lifting scheme and Chaos, Information Sciences, Cilt 520, 177-194.

[67] Çavuşoğlu, Ü. and Kaçar, S., 2019, A novel parallel image encryption algorithm based on Chaos, Cluster Computing, 22(4), 1211-1223.

[68] Wang, X., Xu, M. and Li, Y., 2019, Fast encryption scheme for 3D models based on Chaos System, Multimedia Tools and Applications, 78(23), 33865-33884.

[69] Arab, A., Rostami, M. J. and Ghavami, B., 2019, An image encryption method based on Chaos System and AES algorithm, The Journal of Supercomputing, 75(10), 6663-6682.

[70] Özkaynak, F., 2020, On the effect of chaotic system in performance characteristics of Chaos based S-box designs, Physica A: Statistical Mechanics and its Applications, Cilt 550, 124072.

[71] Babaei, M., 2012, A novel text and image encryption method based on Chaos Theory and DNA computing, Natural Computing, 12(1), 101-107.

[72] Bahrami, M. and Singhal, M., 2015, A light-weight permutation based method for data privacy in Mobile Cloud Computing, 30 March 2015 - 03 April 2015 San Francisco, CA, USA.

[73] Dhall, S., Pal, S. K. and Sharma, K., 2022, A chaos-based probabilistic block cipher for image encryption, Journal of King Saud University - Computer and Information Sciences, 34(1), 1533-1543.

[74] El Assad, S., Farajallah, M. and Vladeanu, C., 2014, Chaos-based block ciphers: An overview, 29-31 May 2014 Bucharest, Romania.

[75] Yasser, I., Mohamed, M. A., Samra, A. S. and Khalifa, F., 2020, A chaotic-based encryption/decryption framework for Secure Multimedia Communications, Entropy, 22(11), 1253.

[76] Fadhil, M. S., Farhan, A. K., Fadhil, M. N. and Al-Saidi, N. M., 2020, A new lightweight AES using a combination of chaotic systems, 12-13 July 2020 Baghdad, Iraq.

[77] Guo, Y., Jing, S., Zhou, Y., Xu, X. and Wei, L., 2020, An image encryption algorithm based on logistic-fibonacci cascade chaos and 3D bit Scrambling, IEEE Access, Cilt 8, 9896-9912.

[78] Hoang, T. M., 2019, A chaos-based image cryptosystem using nonstationary dynamics of logistic map, 16-18 October 2019 Jeju, Korea (South).

[79] Khan, J. S. and Kayhan, S. K., 2021, Chaos and compressive sensing based novel Image Encryption Scheme, Journal of Information Security and Applications, Cilt 58, 102711.

[80] Khan, M. and Jamal, S. S., 2021, Lightweight Chaos-based nonlinear component of block ciphers, Wireless Personal Communications, 120(4), 3017-3034.

[81] Wheeler, D. J. and Needham, R. M., 1995, Tea, a tiny encryption algorithm, Fast Software Encryption, 363-366.

[82] Wheeler, D. J. and Needham, R. M., 1997, Tea extensions. Report, Cambridge University.

[83] Hong, D., Sung, J., Hong, S., Lim, J., Lee, S., Koo, B.-S., Lee, C., Chang, D., Lee, J., Jeong, K., Kim, H., Kim, J. and Chee, S., 2006, Hight: A new block cipher suitable for low-resource device, Lecture Notes in Computer Science, 46-59.

[84] Kumar, M., Pal, S. K. and Panigrahi, A., 2014, FeW: A Lightweight Block Cipher, IACR Cryptol ePrint Arch, 326.

[85] Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B. and Wingers, L., 2015, The simon and Speck lightweight block ciphers, 08-12 June 2015 San Francisco, CA, USA.

[86] Bogdanov, A., Knudsen, L. R., Leander, G., Paar, C., Poschmann, A., Robshaw, M. J., Seurin, Y. and Vikkelsoe, C., 2007, Present: An ultra-lightweight block cipher, Cryptographic Hardware and Embedded Systems - CHES 2007, 450-466.

[87] Zhang, W., Bao, Z., Lin, D., Rijmen, V., Yang, B. and Verbauwhede, I., 2015, Rectangle: A bit-slice lightweight block cipher suitable for multiple platforms, Science China Information Sciences, 58(12), 1-15.

[88] Hong, D., Lee, J.-K., Kim, D.-C., Kwon, D., Ryu, K. H. and Lee, D.-G., 2014, Lea: A 128-bit block cipher for fast encryption on common processors, Information Security Applications, 3-27.

[89] Borghoff, J., Canteaut, A., Güneysu, T., Kavun, E. B., Knezevic, M., Knudsen, L. R., Leander, G., Nikov, V., Paar, C., Rechberger, C., Rombouts, P., Thomsen, S. and Yalçın, T., 2012, Prince – a low-latency block cipher for pervasive computing applications, Advances in Cryptology – ASIACRYPT 2012, 208-225.

[90] Rivest, R. L., 1995, The RC5 encryption algorithm. Fast Software Encryption, 86-96.

[91] Salamia, Y., Khajehvand, V. and Zeinalia, E., 2023, Cryptographic Algorithms: A Review of the Literature, Weaknesses and Open Challenges, Journal of Computer & Robotics, 16(2), 63-115.

[92] Biewald, S., 2024, xtea 0.7.1., https://pypi.org/project/xtea/[Ziyaret tarihi: 20 Mayıs 2024].

[93] Fiefdx, 2024, tea-encrypt 0.0.1., https://pypi.org/project/tea-encrypt/[Ziyaret tarihi: 20 Mayıs 2024].

[94] McCoy, C., 2024, simonspeckciphers 1.0.0., https://pypi.org/project/simonspeckciphers/[Ziyaret tarihi: 21 Mayıs 2024].

[95] Zhuzhuor, 2024, KATAN-Python., https://github.com/bozhu/KATAN-Python[Ziyaret tarihi: 21 Mayıs 2024].

[96] Milczarczyk, S., 2024, Lightweight_Cryptography., https://github.getafreenode.com/szym-i/Lightweight_Cryptography/blob/main/Lightweight_Cryptography_English.ipynb [Ziyaret tarihi: 22 Mayıs 2024].

[97] Sergiichuk, I., 2024, PRESENT-cipher., https://github.com/xSAVIKx/PRESENT-cipher/blob/master/present/pyPresent.py [Ziyaret tarihi: 22 Mayıs 2024].

[98] Eijs, H., 2024, pycryptodome., https://github.com/Legrandin/pycryptodome/blob/v3.11.0/Doc/src/cipher/aes.rst [Ziyaret tarihi: 22 Mayıs 2024].

[99] Whiteman, T., 2024, pyDes., https://github.com/twhiteman/pyDes/blob/master/test_pydes.py [Ziyaret tarihi: 23 Mayıs 2024].

[100] Talirongan, H., Sison, A. M. and Medina, R. P., 2019, Modified Advanced Encryption Standard using Butterfly Effect, 29 November 2018 - 02 December 2018 Baguio City, Philippines.

[101] Shannon, C. E., 1945, A Mathematical Theory of Cryptography, Bell System Technical Memo MM, 45-110-02.

[102] Shannon, C. E., 1949, Communication theory of secrecy systems, Bell System Technical Journal, 28(4), 656-715.

[103] Reddy, B. D., Kumari, V. V. and Raju, K., 2014, A new symmetric probabilistic encryption scheme based on random numbers, 19-20 August 2014 Guntur, India.

[104] Muhammad, K., Hamza, R., Ahmad, J., Lloret, J., Wang, H. and Baik, S. W., 2018, Secure Surveillance Framework for IOT systems using Probabilistic Image Encryption, IEEE Transactions on Industrial Informatics, 14(8), 3679-3689.

[105] Tuncer, T., 2018, A probabilistic image authentication method based on Chaos, Multimedia Tools and Applications, 77(16), 21463-21480.

[106] Papadimitriou, S., Bountis, T., Mavroudi, S. and Bezerianos, A., 2001, A probabilistic symmetric encryption scheme for very fast secure communication based on chaotic systems of difference equations, International Journal of Bifurcation and Chaos, 11(12), 3107-3115.

[107] Leung, K. C., Li, S. L., Cheng, L. M. and Chan, C. K., 2006, A symmetric probabilistic encryption scheme based on CHNN without data expansion, Neural Processing Letters, 24(2), 93-105.

[108] Guo, D. H., Cheng, L. M. and Cheng, L. L., 1999, A new symmetric probabilistic encryption scheme based on chaotic attractors of neural networks, Applied Intelligence, Cilt 10, 71-84.

[109] Kanso, A., Ghebleh, M. and Bou Khuzam, M., 2022, A probabilistic chaotic image encryption scheme, Mathematics, 10(11), 1910.

[110] Puthuparambil, A. B. and Thomas, J. J., 2019, Freestyle, a randomized version of ChaCha for resisting offline brute-force and dictionary attacks, Journal of Information Security and Applications, Cilt 40, 102396.

[111] Hanis, S. and Amutha, R., 2018, A fast double-keyed authenticated image encryption scheme using an improved chaotic map and a butterfly-like structure, Nonlinear Dynamics, 95(1), 421–432.

[112] Yang, H., Wong, K.-W., Liao, X., Zhang, W. and Wei, P., 2010, A fast image encryption and authentication scheme based on chaotic maps, Communications in Nonlinear Science and Numerical Simulation, 15(11), 3507-3517.

[113] Rajput, A. S. and Sharma, M., 2015, A novel image encryption and authentication scheme using chaotic maps, Advances in Intelligent Systems and Computing, 277-286.

[114] Li, C., Luo, G., Qin, K. and Li, C., 2016, An image encryption scheme based on Chaotic Tent Map, Nonlinear Dynamics, 87(1), 127-133.

[115] Hua, Z. and Zhou, Y., 2016, Image encryption using 2D logistic-adjusted-sine map, Information Sciences, Cilt 339, 237-253.

İNTİHAL RAPORU İLK SAYFASI

Yasin KAYA

ORİJİNALLİK RAPORU

%9	%6	%7	%3
BENZERLİK ENDEKSİ	İNTERNET KAYNAKLARI	YAYINLAR	ÖĞRENCİ ÖDEVLERİ

BİRİNCİL KAYNAKLAR

1	Yasin KAYA, Zeynep Gürkaş AYDIN, Akif Akgül. "A chaos-based lightweight encryption scheme using hash-code and cyclic rotation", Physica Scripta, 2025 Yayın	%3
2	Submitted to The Scientific & Technological Research Council of Turkey (TUBITAK) Öğrenci Ödevi	%2
3	acikbilim.yok.gov.tr İnternet Kaynağı	%1
4	hal.archives-ouvertes.fr İnternet Kaynağı	%1
5	Yasin Kaya, Zeynep Gurkas-Aydin, Akif Akgul. "A chaos-based lightweight encryption scheme using hash-code and cyclic rotation", Physica Scripta, 2025 Yayın	<%1
6	dspace.kocaeli.edu.tr:8080 İnternet Kaynağı	<%1
7	hdl.handle.net İnternet Kaynağı	<%1
8	www.researchgate.net İnternet Kaynağı	<%1