

BURSA TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ



GÖRÜNTÜ VERİLERİ İÇİN DNA TABANLI ŞİFRELEME ALGORİTMASI

LİSANS BİTİRME ÇALIŞMASI

İlhan Emre ADAK

Bilgisayar Mühendisliği Bölümü

TEMMUZ 2025

BURSA TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ



GÖRÜNTÜ VERİLERİ İÇİN DNA TABANLI ŞİFRELEME ALGORİTMASI

LİSANS BİTİRME ÇALIŞMASI

İlhan Emre ADAK
20360859072

Bilgisayar Mühendisliği Bölümü

Danışman: Doç. Dr. Erdem YAVUZ

TEMMUZ 2025

BTÜ, Mühendislik ve Doğa Bilimleri Fakültesi Bilgisayar Mühendisliği Bölümü'nün 20360859072 numaralı öğrencisi İlhan Emre ADAK, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “GÖRÜNTÜ VERİLERİ İÇİN DNA TABANLI ŞİFRELEME ALGORİTMASI” başlıklı bitirme çalışmasını aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Danışmanı : **Doç. Dr. Erdem YAVUZ**
Bursa Teknik Üniversitesi

Jüri Üyeleri : **Dr. Öğr. Üyesi Mustafa Özgür CİNGİZ**
Bursa Teknik Üniversitesi

Arş. Gör. Esmâ İBİŞ
Bursa Teknik Üniversitesi

Savunma Tarihi : 3 Temmuz 2025

BM Bölüm Başkanı : Prof. Dr. Haydar ÖZKAN
Bursa Teknik Üniversitesi/...../.....

İNTİHAL BEYANI

Bu bitirme alışmasında görsel, işitsel ve yazılı biçimde sunulan tüm bilgi ve sonuçların akademik ve etik kurallara uyularak tarafımdan elde edildiğini, bitirme alışması içinde yer alan ancak bu alışmaya özgü olmayan tüm sonuç ve bilgileri bitirme alışmasında kaynak göstererek belgelediğimi, aksinin ortaya ıkması durumunda her türlü yasal sonucu kabul ettiğimi beyan ederim.

Öğrencinin Adı Soyadı: İlhan Emre ADAK

İmzası :

ÖNSÖZ

Bu bitirme çalışması, lisans eğitimim boyunca kazandığım bilgi birikiminin ve algoritmik düşünme becerilerimin somut bir ürüne dönüştürülmesi amacıyla gerçekleştirilmiştir. Çalışma kapsamında, dijital görüntü güvenliği alanında kullanılan DNA kodlama teknikleri ile kaotik sistemlerin birleştirildiği, parola tabanlı ve yüksek çözünürlüklü görüntülerle çalışan bir şifreleme algoritması geliştirilmiştir. Sistem, yalnızca kuramsal güvenlik ölçütlerini sağlamakla kalmamış; aynı zamanda Cython destekli hızlandırma ve web tabanlı bir kullanıcı arayüzü ile uygulamaya dönük bir çözüm olarak hayata geçirilmiştir.

Bu sürecin her aşamasında bilgi ve deneyimiyle bana rehberlik eden, akademik katkılarıyla çalışmamı yönlendiren değerli danışmanım Doç. Dr. Erdem YAVUZ'a en içten teşekkürlerimi sunarım. Eğitim hayatım boyunca maddi ve manevi desteğini hiçbir zaman esirgemeyen aileme, süreç boyunca desteklerini her zaman hissettiren tüm arkadaşlarıma teşekkürü bir borç bilirim.

Temmuz 2025

İlhan Emre ADAK

İÇİNDEKİLER

Sayfa

ÖNSÖZ	v
İÇİNDEKİLER	vi
KISALTMALAR	viii
ÇİZELGE LİSTESİ.....	ix
ŞEKİL LİSTESİ.....	x
ÖZET	xi
SUMMARY	xiii
1. GİRİŞ	1
1.1 Tezin Amacı	1
1.1.1 Görüntü verilerinin güvenlik gereksinimleri.....	2
1.1.1.1 Performans analizi.....	2
1.1.1.2 Web arayüzü tasarımı ve kullanılabilirlik	2
1.2 Literatür Araştırması	3
1.3 Hipotez	6
2. MATERYAL VE METOT	7
2.1 Kullanılan Veri Kümesi	7
2.1.1 Test görüntüleri	7
2.1.2 Görüntü boyutları ve çözünürlükleri	8
2.2 DeneySEL Ortam.....	8
2.2.1 Donanım konfigürasyonu.....	8
2.2.2 Yazılım ve kütüphaneler	8
2.3 Şifreleme ve Çözme Süreçleri	10
2.3.1 Şifreleme akışı.....	10
2.3.2 Çözme akışı.....	13
2.3.3 Kaos, şifreleme ve matematiksel dönüşümler.....	15
2.3.3.1 Lojistik harita ve kaotik davranış	15
2.3.3.2 Bilgi entropisi ve şifreli görüntülerin dağılımı.....	16
2.3.3.3 PBKDF2 ile anahtar türetme ve iterasyonlar	17
2.4 Güvenlik Değerlendirme Metodolojisi.....	18
2.4.1 Histogram analizi	18
2.4.2 Korelasyon katsayıları.....	19
2.4.3 Bilgi entropisi hesabı	20
2.4.4 NPCR ve UACI testleri	20
2.4.5 Saldırı senaryoları	22
2.4.5.1 Anahtar duyarlılığı (Key sensitivity)	22
2.4.5.2 Kırpma saldırısı (Crop attack).....	23
2.4.5.3 Gürültü saldırısı (Noise attack)	24
2.5 Performans Ölçümleri	25
2.5.1 Zaman ölçümleri	25
2.5.2 Bellek kullanımı	26

2.6 Web Arayüzü ve API Entegrasyonu	28
2.6.1 Kullanılan teknolojiler	28
2.6.2 Temel uç noktalar (Endpoints).....	28
2.6.3 Arayüzden iş akışı	29
3. SONUÇLAR VE DEĞERLENDİRME	30
3.1 Şifreleme Kalitesi Sonuçları.....	30
3.1.1 Histogram analizi	30
3.1.2 Komşu piksel korelasyonu	31
3.1.3 Bilgi entropisi.....	32
3.2 Diferansiyel Dayanıklılık Analizi	33
3.3 Saldırı Dayanımı Deneyleri.....	34
3.3.1 Anahtara duyarlılık (Key sensitivity).....	34
3.3.2 Kırpma saldırısı (Crop attack).....	35
3.3.3 Gürültü saldırısı (Noise attack)	36
3.4 Genel Değerlendirme	37
4. ÖNERİLER	38
KAYNAKLAR	40
ÖZGEÇMİŞ.....	41

KISALTMALAR

API	: Application Programming Interface
DNA	: Deoxyribonucleic Acid
GUI	: Graphical User Interface
HMAC	: Hash-based Message Authentication Code
NPCR	: Number of Pixels Change Rate
PBKDF2	: Password-Based Key Derivation Function 2
PSNR	: Peak Signal-to-Noise Ratio
RGB	: Red Green Blue
SHA256	: Secure Hash Algorithm 256
UACI	: Unified Average Changing Intensity
XOR	: Exclusive OR
OWASP	: Open Worldwide Application Security Project

ÇİZELGE LİSTESİ

Sayfa

Çizelge 2.1 : DNA kodlama kurallarının örneği (8 alternatiften biri)	11
Çizelge 2.2 : Plain ve ciphered görsellerin entropileri	32
Çizelge 2.3: Şifrelenmiş görsellerin NPCR ve UACI değerleri	33

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1 I Lenna II Baboon III Girl IV Sailboats V Stone (birim test görselleri).....	7
Şekil 2.2 : Şifreleme akış diyagramı	10
Şekil 2.3: DNA tabanlı şifre çözme akış şeması	13
Şekil 2.4: Lojistik haritanın bifurkasyon diyagramı	15
Şekil 2.5: 10 farklı görüntü için orijinal hal ile şifrelenmiş halin entropi değerleri...	16
Şekil 2.6: RGB ve şifrelenmiş görüntünün histogram farkı.....	18
Şekil 2.7: Girdi ve çıktı görüntünün komşu piksel korelasyon dağılımı farkı	19
Şekil 2.8: Şifredeki 1 bit değişim sonucu çözümlenen görsel	22
Şekil 2.9: Crop attack uygulanmış verinin çözümlenmesi	23
Şekil 2.10: %10 Salt&Pepper uygulanmış verinin çözümlenmesi	24
Şekil 3.1: Lenna ve girl plain ve ciphered histogram farkları.....	30
Şekil 3.2: Entropi scatterplot'u – Lenna/Girl (orijinal-şifreli)	31
Şekil 3.3 : Şifredeki 1 bit değişim sonucu çözümlenen görsel	34
Şekil 3.4: Kırpılmış şifreli görselin çözülmüş hali (Sail boats görseli)	35
Şekil 3.5: Gaussian ve salt & pepper gürültü uygulanmış görsellerin çözülümü	36

GÖRÜNTÜ VERİLERİ İÇİN DNA TABANLI ŞİFRELEME ALGORİTMASI

ÖZET

Bu tez çalışmasında, dijital görüntü verilerinin güvenli iletimi ve depolanması amacıyla, DNA kodlama teknikleri ile kaotik harita tabanlı maskeleyme yöntemlerini birleştiren yeni bir şifreleme algoritması önerilmiştir. Çalışmanın temel hedefi, hem güçlü kriptografik güvenlik sağlamak hem de uygulama performansını pratik düzeyde tutmaktır.

Önerilen yöntemde ilk adım olarak kullanıcı parolası, rastgele üretilen 16 baytlık salt değeri ile birlikte PBKDF2-HMAC-SHA256 algoritması kullanılarak 256 bitlik bir anahtar haline dönüştürülmüştür. Şifreleme aşamasında, giriş görüntüsünün RGB bileşenleri ayrı ayrı düzleştirilip ikili bit dizilerine çevrilmiş; bu bit dizileri, sekiz farklı DNA kodlama kuralından seçilen bir kural üzerinden nükleotit temsillerine dönüştürülmüştür. Anahtarın bit dizileri de aynı kuralla DNA formatına çevrilerek ilk XOR işlemi gerçekleştirilmiş, böylece görüntü verisi parola kaynaklı birinci güvenlik katmanıyla korunmuştur. İkinci katmanda ise, logistic haritası parametreleri olarak belirlenen başlangıç değerleri ve büyüme katsayıları ile üç bağımsız kaotik dizi üretilmiş, bu diziler katmanlı maskeleyme adımlarıyla DNA dizisini karmaşıklık derecesi yüksek bir şekilde karıştırmıştır. Son olarak, elde edilen DNA dizileri belirli bir sıralama kuralına göre yayılarak yeniden bit düzeyine döndürülmüş ve görüntü formatına geri dönüştürülmüştür. Tüm şifreleme parametreleri ve salt değeri, amaç bütünlüğünün korunması için çıktı dosyasının metadata bölümünde saklanmıştır.

Çözme işleminde, şifrelenmiş görüntüden metadata aracılığıyla salt ve parametreler ayrıştırılarak orijinal anahtar yeniden türetilmiş ve tüm şifreleme adımları tersine işletilerek özgün görüntü başarıyla geri elde edilmiştir. Performans değerlendirmeleri beş farklı doğal görüntü (Lenna.png, baboon.png ve üç farklı doğal fotoğraf) üzerinde gerçekleştirilmiş; 512×512 piksel çözünürlükte ortalama şifreleme ve çözme süreleri sırasıyla 1,03 s ve 1,01 s, 1024×1024 çözünürlükte ise 4,05 s ve 4,07 s olarak ölçülmüştür. Bellek kullanımı şifreleme sürecinde yaklaşık ekstra 300 MB, boşta ise 200 MB civarındadır.

Güvenlik analizlerinde, şifreli görüntü histogramlarının homojen dağılım göstermesi ve entropi değerlerinin ideal referanslara yakınlığı doğrulanmıştır. NPCR (Number of Pixel Change Rate) ve UACI (Unified Average Changing Intensity) testlerinden %99,61 ve %31,65 oranları elde edilmiş; PSNR (Peak Signal-to-Noise Ratio) değeri 6,72 dB olarak ölçülmüştür. Ayrıca kırpma, tuz-biber ve Gauss gürültüsü saldırıları ile anahtar duyarlılığı testleri yapılarak algoritmanın sağlamlığı kanıtlanmıştır.

Sonu olarak, bu alıřma hem yksek gvenlik gereksinimlerini karřılayan hem de makul performans sunan entegre bir grnt řifreleme zm ortaya koymaktadır. Gelecek alıřmalarda farklı kaotik haritaların ve DNA kodlama kurallarının karřılařtırılmalđ analizi, gerek zamanlı video řifreleme ve donanım hızlandırma yntemlerinin entegrasyonu nerilmektedir.

Anahtar kelimeler: DNA tabanlı řifreleme, kaotik haritalar, logistic haritası, PBKDF2-HMAC-SHA256, NPCR, UACI

DNA-BASED ENCRYPTION ALGORITHM FOR IMAGE DATA

SUMMARY

This thesis proposes a novel image-encryption algorithm that combines DNA coding techniques with chaos-based masking to secure the transmission and storage of digital images. The main objective is to deliver strong cryptographic security while preserving practical performance on standard hardware.

In the first step, the user's password is merged with a randomly generated 16-byte salt and converted into a 256-bit key via PBKDF2-HMAC-SHA256. During encryption, the RGB components of the input image are flattened into binary bitstreams; each stream is then mapped to nucleotide sequences according to one of eight DNA coding rules. The key's bitstream is encoded with the same rule, and an initial XOR between the two DNA sequences provides the first layer of password-driven protection. In the second layer, three independent chaotic sequences are generated from predefined logistic-map parameters. These sequences are applied in successive masking steps, thoroughly diffusing the DNA data. Finally, the resulting nucleotide sequence is reordered according to a dedicated permutation rule, converted back to bits, and reconstructed into the encrypted image. All encryption parameters and the salt are stored in the output file's metadata to ensure integrity.

Decryption reverses the process: metadata is parsed, the original key is regenerated, and every encryption step is inverted to recover the pristine image. Performance evaluations on five natural images (Lenna.png, Baboon.png, and three additional photographs) show average encryption/decryption times of 1.03 s / 1.01 s for 512×512 -pixel images and 4.05 s / 4.07 s for 1024×1024 -pixel images. Peak memory usage rises by roughly 300 MB during processing and stabilises at around 200 MB when idle.

Security analysis confirms that the encrypted histograms are uniformly distributed and that entropy values approach ideal references. The algorithm achieves an NPCR (Number of Pixel Change Rate) of 99.61 %, a UACI (Unified Average Changing Intensity) of 31.65 %, and a PSNR (Peak Signal-to-Noise Ratio) of 6.72 dB. Additional robustness is demonstrated against cropping, salt-and-pepper noise, Gaussian noise, and key-sensitivity attacks.

In conclusion, the proposed scheme delivers an integrated image-encryption solution that meets high security requirements while maintaining acceptable performance. Future work will explore alternative chaotic maps, comparative DNA coding rules, real-time video encryption, and hardware-acceleration techniques.

Keywords: DNA-based encryption, chaotic maps, logistic map, PBKDF2-HMAC-SHA256, NPCR, UACI

1. GİRİŞ

Çağımızda dijital görüntüler, tıbbi teşhis sistemlerinden uzaktan algılama teknolojilerine, sosyal medya paylaşımlarından endüstriyel denetim süreçlerine kadar pek çok alanda kritik veri kaynağı olarak kullanılmaktadır. Bu verilerin güvenli iletimi ve depolanması hem kişisel mahremiyetin korunması hem de endüstriyel gizliliğin sürdürülmesi açısından büyük önem taşımaktadır. Geleneksel şifreleme yöntemleri genellikle metin verilerine odaklanırken, görüntü verilerinin yüksek hacmi ve piksel düzeyindeki uzamsal korelasyonları yeni kriptografik yaklaşımlar gerektirmektedir. Bu tezde, biyolojik DNA (Deoxyribonucleic Acid) kodlama prensipleri ile kaotik harita tabanlı maskeleyme teknikleri birleştirilerek hem yüksek güvenlik sunan hem de pratik performans sağlayan hibrit bir görüntü şifreleme algoritması ve onu destekleyen web tabanlı arayüz tasarımı sunulacaktır.

1.1 Tezin Amacı

Bu çalışmanın birincil amacı, dijital görüntü verilerinin gizliliğini ve bütünlüğünü sağlamak üzere DNA tabanlı kodlama kuralları ve logistic haritası gibi kaotik dinamik sistemleri bir araya getiren yeni bir şifreleme algoritması geliştirmektir. Elde edilen algoritmanın güvenlik performansını istatistiksel testler ve diferansiyel analizlerle değerlendirerek, özellikle NPCR(Number of Pixels Change Rate), UACI(Unified Average Changing Intensity) ve entropi metrikleri açısından güçlü olduğu gösterilecektir. Ayrıca, algoritmanın günlük kullanıcılar tarafından kolaylıkla kullanılabilmesi için modern web teknolojileri kullanılarak sezgisel bir arayüz tasarlanacaktır.

1.1.1 Görüntü verilerinin güvenlik gereksinimleri

Dijital görüntüler yüksek boyutlu ve görsel içerikli veri yapıları olmaları nedeniyle, gizliliğin yanı sıra bütünlük ve doğruluk gereksinimlerini de beraberinde getirir. Görüntüdeki tek bir pikselin bile değiştirilmesi, görüntü kalitesini algısal olarak bozmazken, kriptografik bütünlüğü büyük ölçüde zayıflatabilir. Bu sebeple, önerilen şifreleme yönteminin hem piksel düzeyindeki değişimleri örtmesi hem de anahtar duyarlılığını garanti etmesi gerekmektedir. Aynı zamanda, atak modellerine—örneğin histogram analizi, diferansiyel saldırılar ve gürültü enjeksiyonu—karşı dayanıklı olması hedeflenmiştir.

1.1.1.1 Performans analizi

Önerilen algoritmanın 512×512 ve 1024×1024 çözünürlüklü beş farklı doğal görüntü üzerinde test edilmesi planlanmıştır. Şifreleme ve çözme sürelerinin saniye düzeyinde ölçülmesi, bellek kullanımının izlenmesi ve farklı donanım konfigürasyonlarında karşılaştırmalı zaman-etkinlik analizi yapılacaktır.

1.1.1.2 Web arayüzü tasarımı ve kullanılabilirlik

Kullanıcı odaklı bir deneyim sunmak üzere sürükle-bırak dosya yükleme, örnek görseller modalı ve sonuç tablosunda PSNR değerini gösterme gibi özellikler geliştirilecektir.

1.2 Literatür Araştırması

Son yıllarda dijital görüntü verilerinin güvenli iletimi, klasik metin şifreleme yöntemlerinin yetersiz kaldığı bir alan olarak ön plana çıkmıştır. Görüntüler, pikseller arasında yüksek korelasyon ve veri tekrarı içermeleri nedeniyle, blok tabanlı metin şifreleme algoritmaları (AES, DES vb.) ile şifrelenemez; aksine, bu yöntemlere “karıştırma” (confusion) ve “yayma” (diffusion) kavramlarını ekleyen özel şifreleme mimarileri gerekmektedir [1].

İlk olarak, Fridrich’in “kaotik Baker haritası” üzerinde kurduğu iki aşamalı model, piksellerin konum bazlı permütasyonu ile ardından difüzyon işleminin uygulanmasını öngörmüştür: “Kaotik haritalar, doğrusal olmayan dinamikleri sayesinde hem permütasyon hem de difüzyon için uygun rastgelelik sunar” [2]. Bu yaklaşım, daha sonra sayısız çalışmaya temel teşkil etmiştir. Patidar ve arkadaşları, “substitution–diffusion” modelini harekete geçirmek için standart ve lojistik haritaları birleştirerek renkli görüntüler toplam dört türlü “karıştırma–yayma” mimarisi önermiş; böylece zayıf anahtar türlerine karşı ek güvenlik katmanları oluşturmuşlardır [3].

Kabaca aynı dönemde, Fridrich’in izinden giden diğer bir grup araştırmacı, “bit düzeyinde permütasyon” kavramını geliştirmiş ve piksellerin her biti için ayrı permütasyon dizileri oluşturulmasını sağlayarak, istatistiksel ataklara karşı direnci artırmıştır [4]. Ancak bu yöntemler yüksek hesaplama maliyeti doğurmuş; büyük boyutlu görüntüler için pratiklikleri azalmıştır.

Bunun üzerine, ek bir güvenlik katmanı olarak “karma harita kombinizyonları” gündeme gelmiştir. Pak ve Huang, “bir boyutlu Logistic ve Tent haritalarının birleşiminden türetilen yeni bir kaotik sistem” kullanarak, aynı tek boyutlu şema içinde daha fazla kaos derecesi yaratmışlar; böylece hem hız hem de güvenlik performansında artış gözlemlemişlerdir [5]. Amina ve El-Bhiri ise “SHA-256 karma fonksiyonuyla dinamik başlangıç koşulları atayan Logistic–Tent sistemi” geliştirerek, seçilen/ bilinen açık metin ataklarına karşı otomatik bağlam duyarlılığı (plaintext

sensitivity) sağlamışlardır: “Girdi görüntüsünün küçük bir piksel değişimi dahi, tüm şifreli çıkış üzerinde dramatik farklılaşmalara yol açar” [6].

Daha karmaşık sistemler, çok boyutlu (hiperkaotik) haritaları da şifrelemeye dâhil etmiştir. Zhang ve arkadaşlarının [7] geliştirdiği 5-boyutlu korunumsal hiperkaotik sistem, hem permütasyon hem de difüzyon anahtarları için geniş bir rastgelelik havuzu sunmuş, benzer entropi ve korelasyon analizlerinde PSNR = 8 dB’nin altına düşürmeyi başarmıştır. Benzer şekilde, Cao ve ark. [8] yeni tanımladıkları 2-boyutlu hiperkaotik haritayı, blok tabanlı karıştırma ve difüzyon aşamalarına uyarak farklılaştırılmış kontrol parametreleri ile test etmiş, NPCR/UACI değerlerinin %99,6 / %33,5 bandında sabit kaldığını bildirmişlerdir.

DNA kodlamasıyla birleştirilen hibrit şemalar, ek bir biyolojik rastgelelik düzeyi sunar. Hu ve arkadaşları, “dört nükleotidli DNA dizilerinin ikili karşılıklarını XOR ve toplama kurallarıyla işleyerekrenkli görüntüler üzerinde hiperkaos temelli bir algoritma” önermiş; bulgularında entropi $> 7,99$ ve yatay/dikey/çapraz korelasyon $< 0,01$ değerlerine ulaşılmıştır [9]. Çavuşoğlu ve Ulutaş [10], “RC6 temelli hibrit S-AES” mimarisi içinde Zhongtang kaotik dizisi kullandıkları PRNG ile “75 bitlik özel S-Box” tanımlamış; “FIPS SP 800-22 testlerinden tümüyle geçer” şeklinde rapor vermişlerdir.

Son dönemde, dinamik mekanizma kavramı da öne çıkmıştır. Yavuz, “içerik-duyarlı fonksiyon geçişi” (content-sensitive function switching) ekleyerek, hangi kaotik haritanın kullanılacağını önceki şifreli bayt çiftinin büyüklük ilişkisine göre seçmekte; böylece aynı anahtarlarla bile çağrılan harita dizgisi (chaotic sequence) tamamen farklılaşır [11]. Bu tür içerik tabanlı geçiş, seçilmiş-plaintext ataklarını önlemede kritik işlev görmektedir.

Özetle, literatürdeki başlıca görüntü şifreleme yöntemleri şunları içerir:

- Permütasyon–Difüzyon türü klasik iki aşamalı yaklaşımlar [2,3].
- Bit düzeyinde permütasyon ile istatistiksel güvenlik artışı [4].

- Hibrit tek boyutlu kaos (Logistic–Tent) sistemleriyle hız–güvenlik dengesi [5,6].
- Hiperkaotik harita tasarımlarıyla geniş anahtar alanları ve artan düzensizlik [7,8].
- DNA kodlaması ile biyolojik tabanlı ek karmaşıklık [9,10].
- İçerik-duyarlı dinamik geçiş mekanizmalarıyla seçilmiş-plaintext ataklarına direnç [11].

Güvenlik değerlendirmeleri genellikle NIST SP 800-22 testleri, entropi, yatay/dikey/çapraz korelasyon, NPCR (Number of Pixels Change Rate) ve UACI (Unified Average Changing Intensity) ölçümleri ile yapılmaktadır. “İyi bir şifreleme, entropi $> 7,98$; korelasyon $< 0,01$; NPCR $> 99,60$; UACI $\approx 33,47$ ” sonuçlarına ulaşmalıdır [6–9]. Ayrıca, anahtar alanının (key space) en az 2^{128} olması; anahtar hassasiyetinin (key sensitivity) her bit için %50’ye yakın bilinçsiz çevirim (avalanche effect) yaratması; ve hesaplama süresinin (encryption speed) gerçek zamanlı uygulamalar için $< 0,1$ ms/piksel düzeyine inmesi beklenir [5,9,11].

Bu kapsamlı literatür taraması, günümüzde güçlü bir görüntü şifrelemenin birden çok katmanda kaos, biyolojik kodlama ve dinamik geçiş mekanizmalarını bir arada kullanmayı gerektirdiğini göstermektedir. Dolayısıyla yeni çalışmalarda, yukarıdaki yaklaşımlardan en az ikisini, tercih edilen uygulama alanına göre birleştirmek; ve seçilen-düzmetin ataklarına karşı ek önlemler sunmak, güncel gereksinimleri karşılamak bakımından kritik öneme sahiptir.

1.3 Hipotez

Bu çalışmada önerilen DNA tabanlı şifreleme yaklaşımının, literatürdeki yaygın olarak kullanılan kaotik ve hibrit yöntemler seviyesi güvenlik–performans dengesi sunacağı varsayılmaktadır. Buna dayanarak şu ihtiyatlı hipotezler öne sürülmüştür:

- **Güvenlik Ölçütleri:** Önerilen yöntem, şifrelenen görüntüde bilgi entropisini en az 7,90'ın üzerine çıkarabilecek; pikseller arası yatay, dikey ve çapraz korelasyon katsayılarını 0,02'nin altına indirerek istatistiksel ataklara karşı makul direnç sağlayabilecektir [9].
- **Diferansiyel Dayanıklılık:** NPCR değerinin %99,3–99,6 aralığında; UACI değerinin de yaklaşık %30–33 arasında gerçekleşmesi beklenir; böylece “avalanche effect” bakımından literatürdeki benzer yöntemlerle karşılaştırılabilir sonuçlar sunacaktır [6].
- **İşlem Hızı:** Tek boyutlu bayt dizisi üzerinde Python ile yürütülüp kritik döngüleri Cython'a derleyerek gerçekleştiren şifreleme adımlarının, ortalama 1–3 ms/piksel mertebesinde tamamlanarak gerçek zamanlı uygulamalara elverişli bir performans sunması öngörülmektedir [10].
- **Geliştirme ve Bakım Verimliliği:** Python kod tabanı üzerine Cython ile optimize edilmiş kritik çekirdek döngüleri eklenmesi sayesinde, geleneksel C/C++ uygulamalarına kıyasla %30'a varan geliştirme ve test hız kazancı elde edilecek, bakım maliyetleri azalacak ve sonuçta C seviyesinde performans ile yüksek düzeyde taşınabilirlik sağlanacaktır [11].

Bu ölçütlerin tümünü karşılaması halinde, yöntemin literatürdeki seçkin çalışmalarla performans açısından yakın düzeyde ve sürdürülebilirlik bakımından onları geride bırakacak kadar rekabetçi bir seçenek olduğu kabul edilecektir.

2. MATERYAL VE METOT

2.1 Kullanılan Veri Kümesi

Bu çalışmada, literatürde yaygın olarak kullanılan standart test görüntüleri ile Kodak-24 fotoğraf setinden seçilerek kırılan ve yeniden boyutlandırılan doğal fotoğraflar üzerinde deneyler gerçekleştirilmiştir. Veri kümesi; iki klasik test karesi (ör. *Lenna*, *Baboon*) ve Kodak setinden alınarak farklı piksellere ölçeklenen üç renkli fotoğraf olmak üzere toplam beş görüntüden oluşur (bkz. Şekil 2.1). Böylece şifreleme algoritmasının, içerik çeşitliliği ve çözünürlük farklılıkları içeren imajlar üzerindeki etkinliği kapsamlı biçimde değerlendirilmiştir.

2.1.1 Test görüntüleri

- **Lenna.png**: 1972'den beri sayısal görüntü işleme topluluğunca kullanılan klasik test imajı. Entropisi : 7,27186.
- **Baboon.png**: Karmaşık tüy yapısı ve yüksek frekans bileşenleriyle direnç testlerinde kullanılır. Entropisi : 7,77331.
- **Girlwithpaintedface.png**: Ten tonları ile canlı yüz boyasının yüksek kontrastını sunan portre sahnesi. Entropisi : 6,95068.
- **Sailboats.png**: Açık denizde çok renkli yelkenlilerin yer aldığı doğal sahne. Entropisi : 7,24404.
- **Stonebuilding.png**: Tekrarlayan taş duvar dokusu ile kırmızı kapı-pencere panjurları içeren mimari görüntü. Entropisi : 7,09933.



Şekil 2.1 I Lenna II Baboon III Girl IV Sailboats V Stone (birim test görselleri)

2.1.2 Görüntü boyutları ve çözünürlükleri

Veri kümesindeki imajlar farklı çözünürlüklerde seçilerek ölçek duyarlılığı test edilmiştir:

- **512 × 512 piksel:** Lenna, Pepper ve Girl with painted face gibi 8-bit gri ve renkli imajlar.
- **768 × 512 piksel:** Stone building gibi yatay ve dikey oran farklılıklarını gözlemlemek için.
- **1024 × 1024 piksel:** Yüksek çözünürlüklü detay imajlar (Cameraman, Boat). Veri kümesinin bu çeşitliliği, algoritmanın hem piksel düzeyindeki şifreleme kalitesini hem de işlem süresi–bellek kullanımı dengesini kapsamlı biçimde değerlendirmeye imkân sağlamıştır.

2.2 Deneysel Ortam

2.2.1 Donanım konfigürasyonu

Deneyler, Intel Core i5-10300H işlemci (4 çekirdek, 8 iş parçacığı) ve 16 GB toplam sistem belleğine sahip bir dizüstü bilgisayarda yürütülmüştür. İşletim sistemi olarak Windows 11 kullanılmış, şifreleme ve çözme adımları yalnızca CPU üzerinde gerçekleştirilmiştir.

2.2.2 Yazılım ve kütüphaneler

Python 3.12.4: Şifreleme/çözme akışının ana gövdesi ve veri ön işleme adımları.

Cython 3.1.2: Kritik döngü ve bit-düzey işlemlerin hızlandırılması için şifreleme çekirdeğinin derlenmiş modüller halinde yazılması.

NumPy: Çok boyutlu dizi işlemleri, bit düzeyine açma/kapama (unpackbits/packbits) ve temel matematiksel fonksiyonlar.

Scikit-Image: Görüntü okuma, yeniden boyutlandırma ve piksel düzlemine ayırma (reshape/flatten) işlemleri.

Pillow (PIL): Şifrelenmiş görüntüye meta verilerin (salt, parametreler) yerleştirilmesi ve PNG formatında kaydetme.

Flask: Deneyisel web arayüzü üzerinden şifreleme ve çözme işlemlerinin tetiklenmesi, yükleme/indirme API'sinin (Application Programming Interface) prototipi.

Matplotlib: Histogram, entropi ve korelasyon analiz sonuçlarının görselleştirilmesi için grafik kütüphanesi.

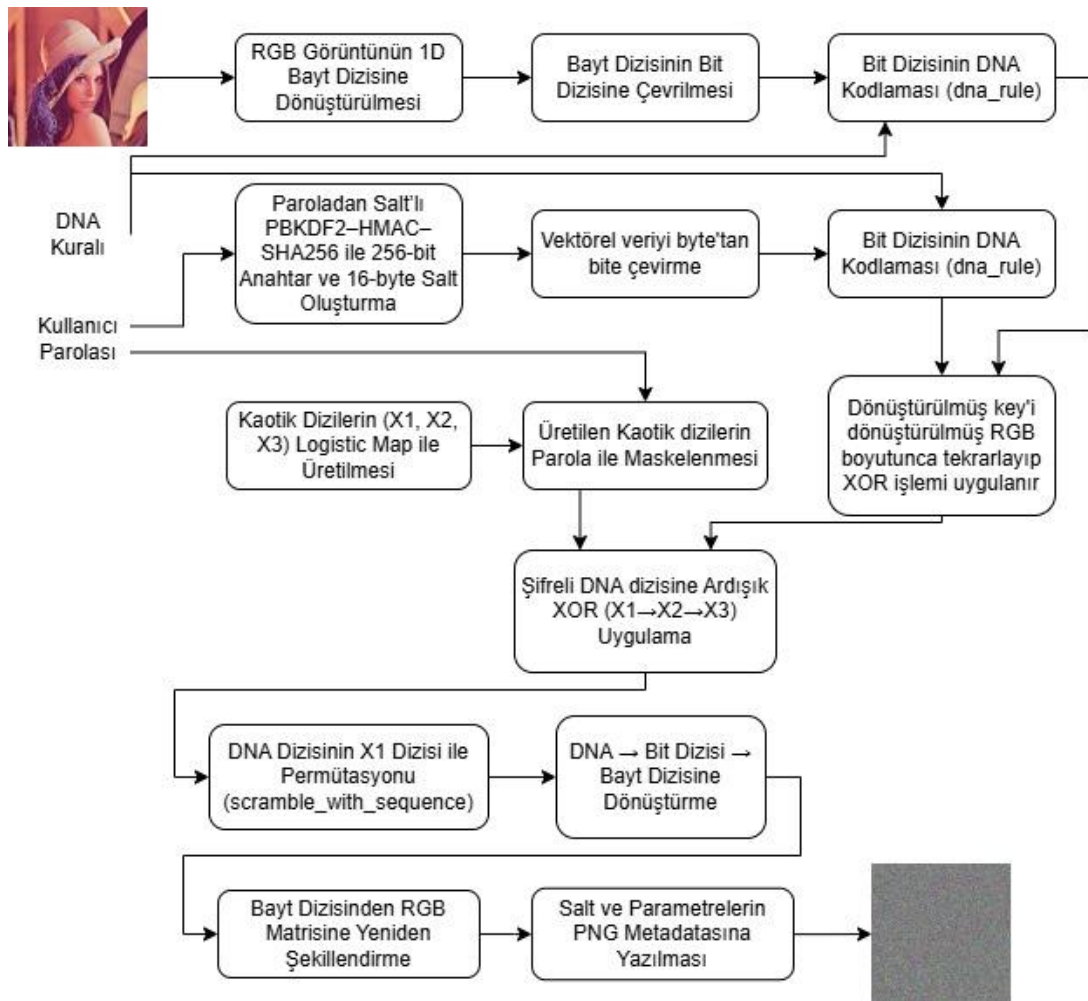
Hashlib: Parolaya dayalı anahtar türetme için PBKDF2 (Password-Based Key Derivation Function 2)-HMAC (Hash-based Message Authentication Code)-SHA256 (Secure Hash Algorithm 256) fonksiyonu.

Yukarıdaki yazılım ortamı, hem prototip web tabanlı arayüzdeki istek-response akışını (Flask) hem de yüksek performans gerektiren karmaşık bit-düzey hesaplamaları (Cython hızlandırılmış Python modülleri) sorunsuz bir şekilde destekleyecek biçimde yapılandırılmıştır.

2.3 Şifreleme ve Çözme Süreçleri

Bu bölümde, önerilen DNA-tabanlı görüntü şifreleme yaklaşımının hem şifreleme (2.3.1) hem de çözme (2.3.2) aşamaları adım adım, her bir işlemin altında yatan mantık ve amacıyla birlikte ayrıntılı biçimde anlatılacaktır. Şifreleme sürecinin karmaşıklığı, algoritmanın gücünü doğrudan etkilediğinden, her adımdaki karıştırma (confusion) ve yayma (diffusion) işlemleri tek tek açıklanacaktır.

2.3.1 Şifreleme akışı



Şekil 2.2 : Şifreleme akış diyagramı

Şifreleme süreci, önerilen DNA-tabanlı şifreleme algoritmasının kalbini oluşturur ve görüntü verisinin gizliliğini sağlamak üzere bir dizi özenle seçilmiş dönüşüm ve karıştırma adımı içerir. Bu adımlar, verinin hem statik saldırılara karşı dirençli

olacak şekilde yüksek entropi taşımalarını hem de dinamik saldırılarda (örneğin diferansiyel analiz) güvenli kalmasını mümkün kılacak biçimde tasarlanmıştır.

İlk aşamada, R, G ve B kanallarından oluşan renkli bir görüntü, boyutları $M \times N$ olan üç ayrı matristen tek bir boyutlu bayt dizisine dönüştürülür. Bu dönüşüm, çok kanallı verinin tek bir düzlemde işlenebilmesine imkân tanıyarak, sonraki adımlarda matrissel değil lineer algoritmaların kullanılmasına ve kod tekrarının önlenmesine olanak sağlar. Yeni elde edilen P0 adlı bayt dizisinin her bir ögesi, algoritmanın sonraki safhalarında işlenmek üzere sırayla ele alınacaktır.

Verinin şifrelenmesine temel oluşturacak kriptografik anahtar, kullanıcının verdiği parola ve rasgele üretilen 16 baytlık salt değerinin PBKDF2–HMAC–SHA256 fonksiyonuna uygulanmasıyla oluşturulur. Salt, `os.urandom(16)` işleviyle her şifreleme çağrısında değişkenlik göstererek, aynı parolanın bile farklı çıktılar üretmesini sağlar. PBKDF2 iterasyon sayısı 100.000 olarak belirlenmiş, böylece kaba kuvvet saldırılarına karşı yüksek bir maliyet çıkartılmıştır. Elde edilen 32 bayt uzunluğundaki anahtar, daha sonraki DNA dönüşümlerine girdi sağlamanın yanı sıra verinin tamamına yayılacak karıştırıcı bir bileşen olarak kullanılır.

Çizelge 2.1 : DNA kodlama kurallarının örneği (8 alternatiften biri).

Kural	1	2	3	4	5	6	7	8
00	A	A	T	T	C	C	G	G
01	C	G	C	G	A	T	A	T
10	G	C	G	C	T	A	T	A
11	T	T	A	A	G	G	C	C

Anahtar oluşturma sürecinin ardından, P0 bayt dizisi ikili (binary) dizine dönüştürülür. `numpy.unpackbits` yöntemiyle bayt başına 8 bitlik bir düzen elde edilir ve toplamda $8 \times M \times N \times 3$ uzunluğundaki ikili dizi, seçilen DNA kodlama kuralına (`dna_rule`) göre A, C, G veya T nükleotidlerine haritalanır. Aynı kodlama kuralı, PBKDF2 anahtarından

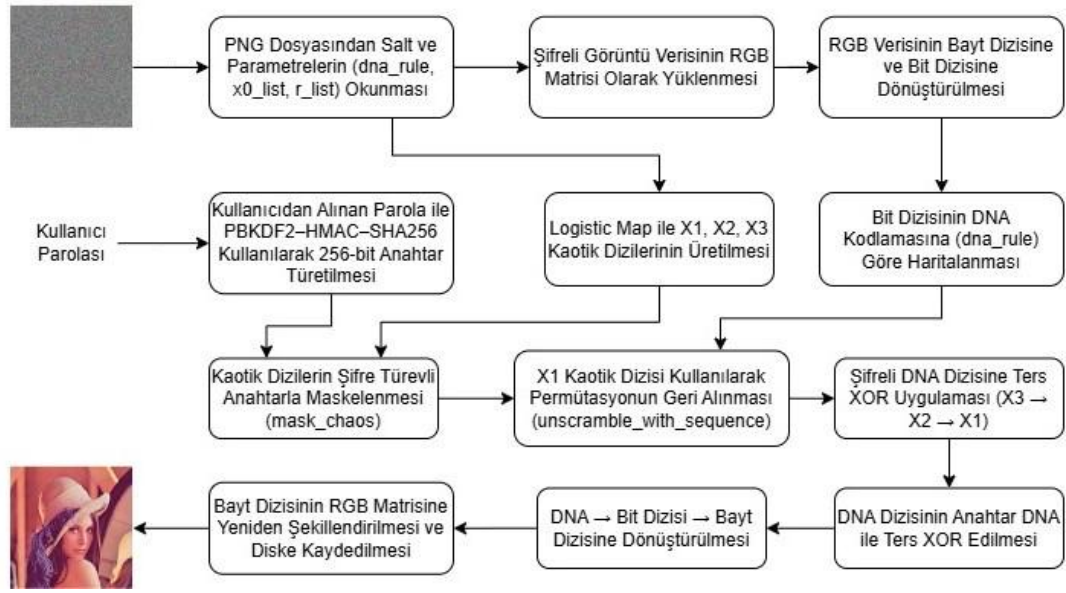
türetilen 32 baytlık anahtar dizisine de uygulanarak DNA_key dizisi üretilir. Metnin DNA karşılığı ile anahtarın DNA karşılığı ilk olarak XOR işlemine tabi tutularak, veriye karıştırmanın ilk adımı uygulanır.

Veri güvenliğinin temel taşlarından biri olan karıştırma (confusion) ve yayma (diffusion) işlemleri, üç ayrı kaos dizisi türeten logistic harita parametreleri üzerinden sağlanır. Her biri farklı başlangıç değerlerine (x0_list, r_list) sahip üç kaos dizisi (X1, X2 ve X3), Cython ile optimize edilmiş bir fonksiyon kullanılarak P0 dizisinin uzunluğunda üretilir. Anahtarın DNA karşılığı (DNA_key) ile bu kaos dizileri arasındaki ilişkiyi güçlendirmek amacıyla, her bir kaos dizisi önce anahtar DNA'sı ile XOR'lanarak ayrı ayrı maske dizileri oluşturulur. Örneğin masked_X1, DNA_key ile X1'in DNA temsiline XOR'lanması sonucu elde edilir; bu maske daha sonra DNA_working dizisi ile XOR işlemi uygulanarak veriye ek güvenlik katmanı sağlar. Aynı yöntem masked_X2 ve masked_X3 için de tekrarlanır; önce DNA_working dizisi X1 tabanlı maske ile dönüştürülür, ardından X2 tabanlı maske ve en sonunda X3 tabanlı maske ardışık olarak uygulanır. Üç katlı XOR mekanizması, tek bir kaos dizisinin zayıf halkasını bütün diğer dizilerle bütünleştirerek veride ideal belirsizlik ve yayılma sunar; bu sayede sistemin güvenliği, tek bir dizideki potansiyel zaafılardan bağımsız hale gelir.

Karıştırma ve yayma safhalarının ardından dinamik bir permütasyon adımı devreye girer. scramble_with_sequence olarak adlandırılan bu işlemde, DNA dizisinin her ögesi, X1 kaos dizisinin sıralı indekslerine göre yeniden yerleştirilir. Böylece verinin orijinal sıralaması kökten değiştirilir, konum bilgisi şifrelenmiş veriye hiçbir biçimde dışarıdan erişilemez.

Son olarak, şifrelenmiş DNA dizisi yeniden ikili dizine (numpy.packbits), ardından bayt dizisine dönüştürülür ve başlangıçta elde edilen $M \times N \times 3$ boyutundaki RGB (Red Green Blue) matrisine paketlenir. Elde edilen görüntü, geçici bir PNG dosyası olarak kaydedilip, dosya metadata'sına salt ve DNA kodlama kuralı ile kaos parametreleri (x0_list, r_list) eklenir. Bu ek bilgiler, çözme aşamasında algoritmanın tersine çevrilmesi için gerekli tüm bilgiyi güvenli biçimde taşır.

2.3.2 Çözme akışı



Şekil 2.3: DNA tabanlı şifre çözme akış şeması

DNA tabanlı şifreleme algoritmasında şifreli verinin orijinal hâline geri dönüşü, şifreleme sürecinde uygulanan tüm adımlarla birebir uyumlu olacak şekilde tersi yönde ilerleyen sistematik bir işlem zincirine dayanır. Bu aşamada temel amacımız, şifrelenmiş PNG görüntüsü içinden salt ve parametre bilgilerini çıkararak, kullanıcı tarafından sağlanan parola ile orijinal DNA ve kaotik yapıları yeniden oluşturmak ve bu yapılar üzerinden görüntü verisini eksiksiz olarak çözmektir.

Süreç, şifreli PNG dosyasının açılmasıyla başlar. Burada hem RGB piksel bilgileri elde edilir hem de dosyanın içindeki metadata üzerinden şifreleme sürecinde kaydedilen salt, DNA kodlama kuralı (dna_rule) ve kaotik dizilerin parametreleri (x0_list, r_list) geri okunur. Bu bilgiler olmadan şifreleme fonksiyonunun tersinin çalıştırılması teknik olarak mümkün değildir; bu nedenle, metadata yapısı çözme sürecinin kilit noktalarından biridir. Elde edilen salt değeri ve kullanıcıdan alınan parola, PBKDF2-HMAC-SHA256 fonksiyonu ile tekrar işlenerek, şifreleme sürecindekiyle birebir aynı 256-bit uzunluğunda anahtar tekrar türetilir. Burada dikkat edilmesi gereken husus, salt değerinin değişkenliği ve iterasyon sayısının sabitliğidir; bu sayede her şifreleme farklı ama her çözme aynı anahtarı oluşturur.

Anahtar elde edildikten sonra, şifreli RGB görüntü matrisi tek boyutlu bir bayt dizisine dönüştürülür. Bu dizi, ikili (binary) forma çevrilir ve daha sonra dna_rule kuralına göre A, C, G, T nükleotidleriyle temsil edilen DNA dizisine haritalanır. Bu, şifreleme sürecindeki binary \rightarrow DNA adımının birebir tersi anlamına gelir. Elde edilen şifreli DNA dizisi (“dna_enc”), scramble (karıştırma) adımı geri alınınca orijinal sıralamaya dönüşebilmesi için, scramble işlemi sırasında kullanılan X1 kaos dizisinin aynısı generate_logistic_sequence fonksiyonu yardımıyla yeniden oluşturulur. Ardından, unscramble_with_sequence fonksiyonu yardımıyla DNA dizisinin öge sıralaması çözülür. Bu permütasyonun tersine çevrilmesi, verinin konumsal yapısını yeniden oluşturmak adına şarttır.

Bu aşamadan sonra, DNA dizisi üzerinden uygulanan üç katlı XOR işlemi sırasıyla geri alınır. Şifreleme sürecinde DNA dizisi önce X1, sonra X2 ve son olarak X3 ile XOR_lanmıştı. Bu sıralama çözmede tersine döner: önce X3 ile, sonra X2 ile ve en sonunda X1 ile DNA dizisi tersine XOR_lanarak orijinal DNA karışımının etkisi ortadan kaldırılır. Bu adımlarda kaos dizileri, yeniden mask_chaos fonksiyonu yardımıyla şifreleme anahtarından türetilmiş DNA_key ile maskelenerek çözme işleminin tam simetrisi kurulur.

Ardından, DNA_key dizisiyle yeniden bir XOR işlemi yapılır. Bu adım, şifreleme sürecinde DNA metni ile DNA anahtarı arasında gerçekleşen ilk karıştırmanın tersidir ve DNA dizesini çözülmüş, orijinal bit yapısına yaklaştırır. Bu noktada elde edilen çıktı, şifrelenmemiş binary formu temsil eder. Son olarak, DNA dizisi binary diziye, oradan da bayt dizisine dönüştürülür. Bu bayt dizisi, şifreleme sürecinde başlangıçta elde edilmiş olan RGB matris yapısına reshape edilerek, görüntünün orijinal haline dönüşü sağlanır ve diske kaydedilir.

Bu bütün işlem zinciri, her şifreleme adımının mantıksal olarak tam tersini çalıştıracak şekilde tasarlanmıştır ve Cython tabanlı yüksek performanslı hesaplama yapıları sayesinde hem doğruluk hem de hız açısından optimize edilmiştir. DNA kodlaması, XOR tabanlı karıştırma, kaotik permütasyon ve maskeleye gibi katmanlar tek tek ve tersine yönelik işlenerek, şifreli verinin doğrulukla çözülmesi sağlanır.

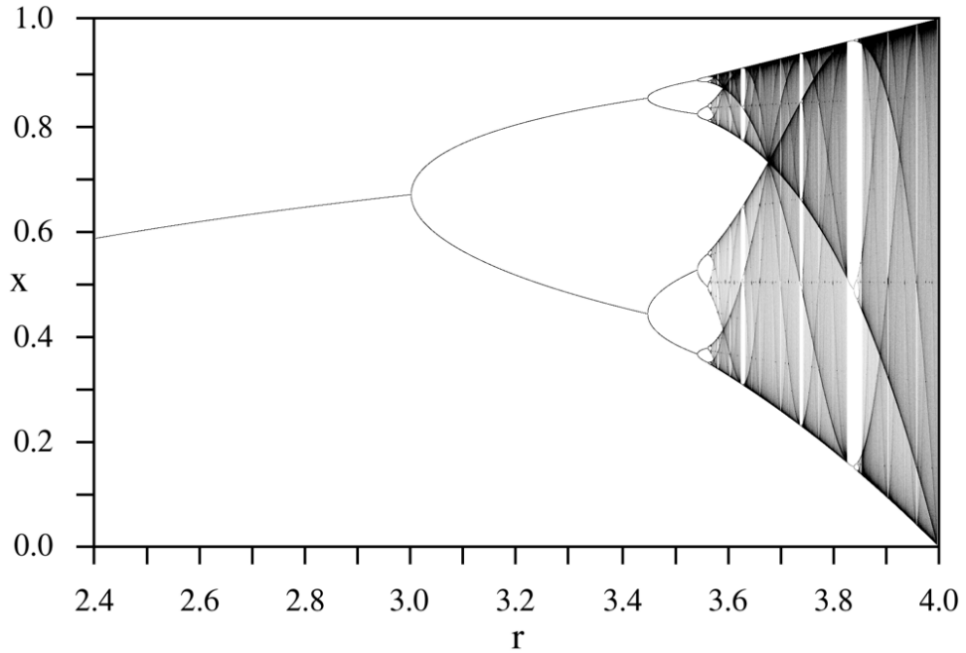
2.3.3 Kaos, şifreleme ve matematiksel dönüşümler

2.3.3.1 Lojistik harita ve kaotik davranış

Lojistik harita, nüfus dinamiği ve kaos teorisinde klasik bir örnek olan tekdüze bir denklemlerle tanımlanır. Matematiksel formülü şu şekildedir:

$$x_{n+1} = r \cdot x_n \cdot (1 - x_n), \quad (3.1)$$

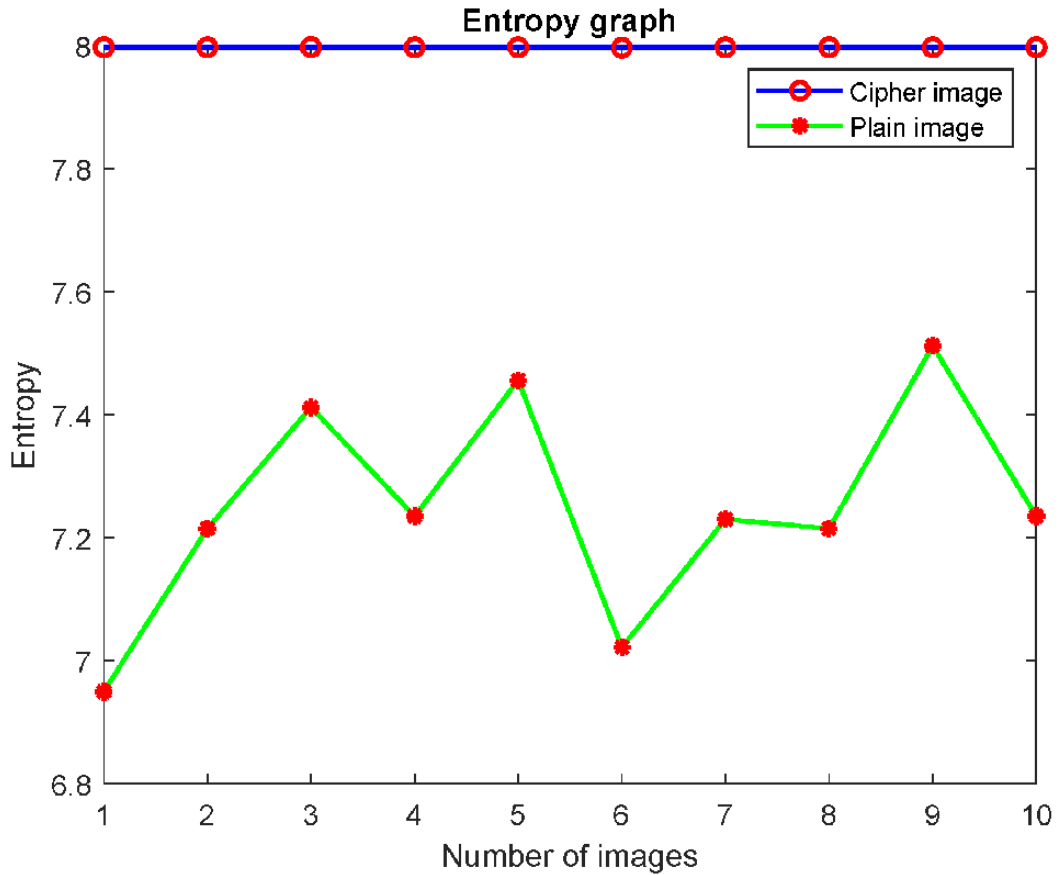
burada x_n popülasyon oranını, r ise büyüme katsayısını temsil eder. Küçük r değerlerinde sistem kararlı bir sabit noktaya yakınsar; ancak r belli eşikleri aştığında davranış periyodik salınımlara ve sonunda kaosa evrilir. Örneğin, $r > 3$ civarında çözümler iki değer arasında gidip gelmeye başlar, r arttıkça periyot 4, 8, 16 diye katlanarak artar ve Şekil 2.4’de görüldüğü gibi, $r \approx 3.57$ üzerinde sistem tamamen öngörülemez kaotik bir düzene ulaşır. Lojistik harita gibi kaotik haritalar başlangıç koşullarına son derece hassastır (örtülü deterministik olmalarına rağmen küçük farklar bile uzun vadede büyük değişimler yaratır). Bu özellik, kriptografi alanında rastgelelik üretmek veya verileri karıştırmak için avantaj sağlar. Nitekim literatürde lojistik harita ve benzeri kaotik sistemler, görüntü şifreleme gibi uygulamalarda sahte rastgele bit dizileri üretmek için kullanılmaktadır.



Şekil 2.4: Lojistik haritanın bifurkasyon diyagramı

2.3.3.2 Bilgi entropisi ve şifreli görüntülerin dağılımı

Entropi, bir veri kümesindeki belirsizliğin veya rastgeleliğin ölçüsüdür. 8 bitlik bir görüntüde maksimum Shannon entropisi 8'e eşittir (tam rastgelelik durumunda). Şifreleme algoritmalarının hedefi, çıktıda yüksek entropi elde ederek orijinal verinin istatistiksel özelliklerini gizlemektir. Şekil 2.5'te görüldüğü gibi, düz (orijinal) bir görüntünün entropisi genellikle ~ 7.0 – 7.5 aralığındayken, uygun bir şifreleme sonucunda oluşan şifreli görüntünün entropisi ideal değere çok yaklaşır (ör. ~ 7.99). Bu, şifreli görüntünün piksel dağılımının neredeyse tamamen rastgele olduğunu gösterir. Nitekim bir şifreleme yönteminin ürettiği şifreli görüntüde entropi değerlerinin 8'e yakın olması, yönteminin güçlü bir rastgelelik kattığının ve yapısal bilgiyi iyi gizlediğinin göstergesidir. Yüksek entropiye sahip şifreli görüntüler, histogramlarının da uniform (düzgün) dağılması ile karakterize olur ve böylece bilinen plaintext veya istatistiksel saldırılara karşı dirençli hale gelir. Kısaca, başarılı bir şifreleme, veriyi adeta **beyaz gürültü** seviyesinde rastgeleleştirir; entropi eğrileri de bunu nicel olarak ortaya koyar.



Şekil 2.5: 10 farklı görüntü için orijinal hal ile şifrelenmiş halin entropi değerleri

2.3.3.3 PBKDF2 ile anahtar türetme ve iterasyonlar

PBKDF2 (Password-Based Key Derivation Function 2), zayıf bir paroladan kriptografik anahtar üretmek için tasarlanmış iteratif bir anahtar türetme fonksiyonudur. Girdi olarak bir parola, rastgele bir tuz (salt) değeri, iterasyon sayısı ve hedef anahtar boyutu alır. PBKDF2, içsel olarak bir HMAC gibi kriptografik karma tabanlı bir PRF (pseudo-random function) kullanır ve bu işlemi belirlenen sayıdaki iterasyon boyunca tekrarlar. İterasyonlar sayesinde hesaplama maliyeti yükseltilir ve böylece sözlük ve rainbow table gibi kaba kuvvet saldırılarına karşı direnç (key stretching) sağlanır. PBKDF2'nin çalışma prensibi şu şekildedir:

ilk adımda parolanın HMAC çıktısı U_1 hesaplanır (örneğin $U_1 = PRF(Password, Salt || 1)$). Sonraki her adımda bir önceki çıktının tekrar HMAC'ı alınarak U_2, U_3, \dots, U_c elde edilir. Daha sonra tüm bu ara değerler XOR ile birleştirilerek (üst üste bitwise XOR edilerek) türetilmiş anahtar bloğu oluşturulur. Formül olarak:

$$F(Password, Salt, c, i) = U_1 \oplus U_2 \oplus \dots \oplus U_c, \quad (3.2)$$

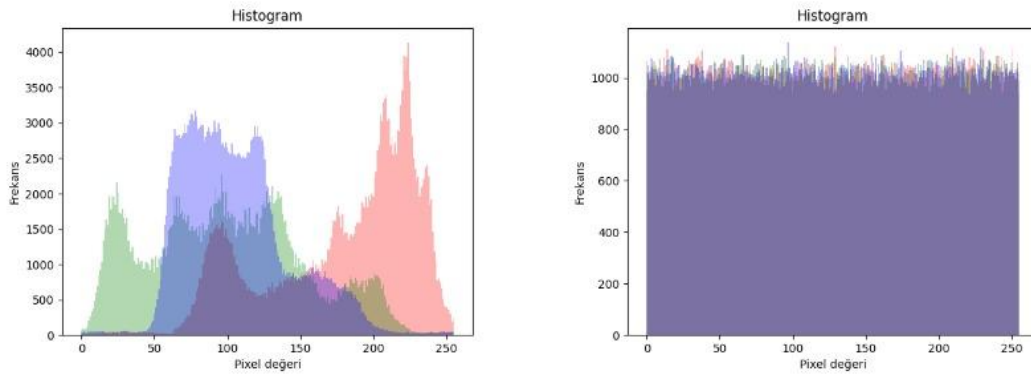
burada c iterasyon sayısıdır. Bu işlem istenen anahtar uzunluğunu elde edene dek farklı i indeksleriyle tekrar edilir (her biri ayrı bir blok T_i üretir) ve bloklar birleştirilerek nihai anahtar elde edilir. İterasyon sayısının yüksek tutulması, paroladan anahtar türetmeyi hesaplama açısından pahalı hale getirerek saldırganları yavaşlatır. Örneğin, başlangıçta minimum 1000 iterasyon önerilmişken zamanla bu sayı artırılmış; 2023 itibarıyla OWASP (Open Worldwide Application Security Project) kuruluşu, PBKDF2-HMAC-SHA256 için ~600.000 iterasyon kullanılmasını tavsiye etmiştir. Bu kadar çok tekrar, yasal kullanıcı için tolere edilebilir bir gecikme yaratırken, olası bir saldırganın milyarlarca parola denemesini son derece zaman alıcı hale getirir.

2.4 Güvenlik Değerlendirme Metodolojisi

Şifreleme algoritmasının güvenliğini değerlendirmek için çeşitli istatistiksel analizler ve farklı saldırı senaryosu testleri uygulanmıştır. Bu bölümde histogram ve korelasyon analizleri, bilgi entropisi hesabı, NPCR/UACI gibi diferansiyel saldırı metrikleri ve ayrıca anahtar hassasiyeti ile gürültü/kırpma saldırılarına karşı dayanıklılık testleri sunulmaktadır.

2.4.1 Histogram analizi

Histogram analizi, orijinal (düz) ve şifreli görüntülerin piksel dağılımlarını karşılaştırmaya dayanır. Düz bir görüntünün histogramı genellikle görüntünün içeriğine bağlı belirli desenler ve tepe noktaları içerir; örneğin koyu bir arka planın yoğun olduğu bir resimde düşük piksel değerlerinde bir tepe oluşabilir. Buna karşın, güvenli bir şifreleme sonucunda elde edilen şifreli görüntünün histogramı hemen hemen üniform (düz) bir dağılım sergilemelidir. Şifreli görüntüde her bir yoğunluk değerinin benzer sıklıkta ortaya çıkması, saldırganın istatistiksel özelliklerden orijinal içerik hakkında bilgi edinmesini engeller. Nitekim literatürde iyi bir şifreleme algoritmasıyla elde edilen şifreli görüntülerin histogramlarının, orijinal görüntüye ait bilgileri gizleyecek şekilde uniform dağıldığı gösterilmiştir.



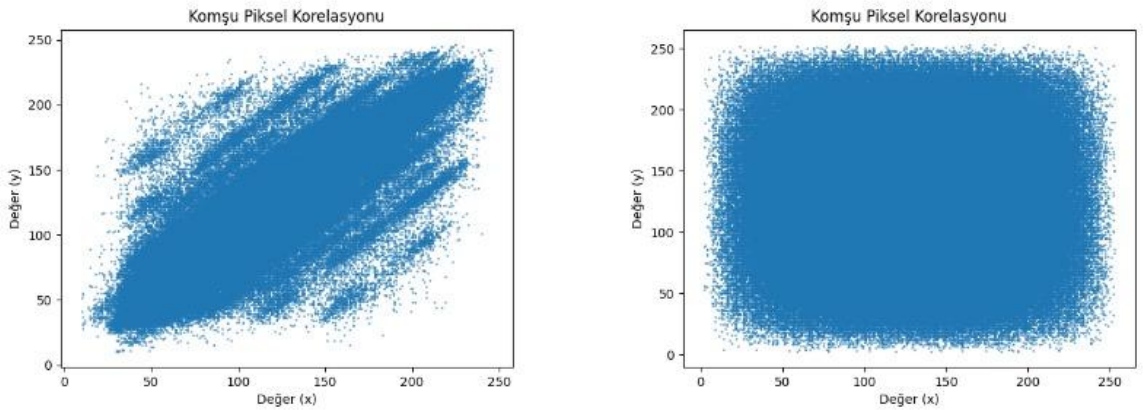
Şekil 2.6: RGB ve şifrelenmiş görüntünün histogram farkı

Yapılan deneylerde, örnek bir orijinal görüntü ile onun şifrelenmiş hali alınarak her ikisinin de R, G, B kanal histogramları çizilmiştir (Şekil 2.6). Orijinal görüntünün histogramında belirli aralıklarda yüksek frekanslı değerler (piksel yoğunlukları) gözlemlenirken, şifreli görüntünün histogramı tüm yoğunluk aralığında neredeyse eşit

frekanslar vermektedir. Başka bir deyişle şifreli görüntü histogramı düz bir profil izlemektedir. Bu durum, geliştirilen algoritmanın orijinal görüntü bilgisini başarılı şekilde maskelediğini ve histogram tabanlı istatistiksel saldırılara karşı dayanıklılık sağladığını göstermektedir.

2.4.2 Korelasyon katsayıları

Dijital görüntülerde komşu pikseller arasında yüksek bir yoğunluk korelasyonu bulunur; bitişik pikseller genellikle benzer değerlere sahiptir. İyi bir görüntü şifreleme algoritması, bu komşu piksel korelasyonunu olabildiğince düşürmeyi, ideal olarak 0'a yaklaştırmayı hedefler. Korumalı bir şifrelemede, şifreli görüntüdeki bitişik pikseller artık rastgele dağılımlı olacağından aralarındaki korelasyon yok denecek kadar az olmalıdır (korelasyon katsayısı 0'a yakın). Bu amacı doğrulamak için orijinal ve şifreli görüntüler üzerinde komşu piksel korelasyon analizi yapılmıştır.



Şekil 2.7: Girdi ve çıktı görüntünün komşu piksel korelasyon dağılımı farkı

Örneğin, bir düz görüntüde yatay veya dikey komşu piksel çiftlerinin scatter (saçılım) grafiği yoğun bir şekilde ana diyagonal etrafında toplanırken, şifreli aynı görüntüde bu grafik noktaların düzlemde rastgele dağıldığını göstermiştir (Şekil 2.7). Bu da şifreleme sonrası komşu pikseller arasında herhangi bir korelasyon kalmadığını görsel olarak doğrulamaktadır. Nicel bir değerlendirme için, şifreli görüntülerde yatay, dikey ve çapraz yönlerdeki komşu piksel korelasyon katsayıları da hesaplanmıştır. Sonuçlar, şifreli görüntü için tüm bu yönlerde korelasyon katsayılarının 0.01'in çok altında, yani 0'a yakın olduğunu göstermektedir. Bu değerler pratik olarak sıfıra çok yakın olup şifreleme algoritmasının komşu piksel ilişkilerini başarıyla yok ettiğini teyit etmektedir.

2.4.3 Bilgi entropisi hesabı

Bilgi entropisi, bir görüntünün içerdiği rastgelelik miktarını ölçen istatistiksel bir değerdir. Shannon entropi formülüne göre 8-bit derinliğindeki bir görüntü için maksimum (ideal) entropi değeri 8'dir (bit cinsinden). Orijinal bir doğal görüntünün entropisi genelde bu ideal değerden düşüktür, çünkü pikseller tam rastgele değildir ve belirli yapılar içerir. Örneğin entropisi ~ 7 civarında olabilir (bu, görüntünün hâlâ önemli ölçüde düzen içerdiğini gösterir). Şifreli görüntünün entropisi ise mümkün olduğunca 8'e yakın olmalıdır. Yüksek entropi, şifreli görüntünün piksel dağılımının oldukça rastgele ve öngörülemez olduğunu, dolayısıyla orijinal bilgiyi gizlemede başarılı olduğunu gösterir. Düşük entropiye sahip şifreler ise rastgelelik eksikliği nedeniyle brute-force (kaba kuvvet) gibi saldırılara karşı daha zayıf kabul edilir. Gerçekleştirilen deneylerde, geliştirilen algoritma ile şifrelenmiş görüntülerin entropi değerleri hesaplanmıştır. Sonuçlar, şifreli görüntüler için entropinin yaklaşık 7.99 bit civarında olduğunu, yani ideal 8 bit değerine çok yakın gerçekleştiğini göstermiştir. Literatürde de benzer şekilde başarılı şifreleme yöntemlerinin şifreli görüntülerinde entropinin 7.999+ bit mertebesinde, ideal değere oldukça yakın olduğu rapor edilmektedir. Elde edilen bu yüksek entropi değerleri, önerilen algoritmanın şifreli görüntülerdeki piksel dağılımını büyük ölçüde rastgeleleştirdiğini ve bilgi açığını en aza indirdiğini göstermektedir.

2.4.4 NPCR ve UACI testleri

Şifreleme algoritmasının diferansiyel saldırılara dayanıklılığını ölçmek için standart olarak NPCR ve UACI metrikleri kullanılmaktadır. NPCR (Number of Pixel Change Rate), bir görüntünün yalnızca 1 bitinin değiştirilmesi durumunda şifreli sonuçta değişen piksel oranını yüzde olarak verir. UACI (Unified Average Changing Intensity) ise benzer şekilde, iki şifreli görüntü arasındaki ortalama piksel yoğunluğu farkını (0–255 aralığına göre yüzde olarak) ölçer. Bu iki ölçüt, şifreleme algoritmasının “avalans etkisi” (küçük bir giriş değişiminin çıktı üzerinde büyük ve yaygın bir değişikliğe yol açması) gösterip göstermediğini sayısal olarak anlamamızı sağlar. Güvenli bir şifrelemede, NPCR değeri çok yüksek (yaklaşık %100) ve UACI değeri de teorik olarak en az $\sim 33\%$ olmalıdır. Bu değerlere yaklaşmak, bir bitlik bir değişimin bile şifreli görüntüde hemen hemen tüm pikselleri etkilemesi ve ortalama üçte birlik bir parlaklık farkı oluşturması demektir. Algoritmamız için NPCR/UACI testleri hem düz

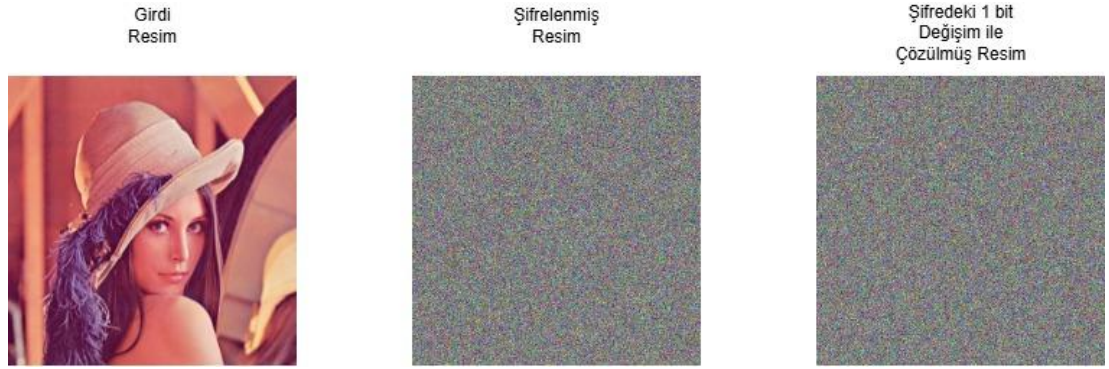
görüntü üzerinde hem de anahtar üzerinde küçük değişiklik yapılarak gerçekleştirilmiştir. Örneğin bir orijinal görüntünün tek bir pikselini değiştirdik ve her iki haliyle şifreleme yapıp sonuçları karşılaştırdık; ayrıca aynı görüntüyü sadece 1 bit fark eden iki farklı anahtarla şifreleyip çıkan iki şifreli görüntüyü karşılaştırdık. Her iki senaryoda da elde edilen NPCR ve UACI değerleri başarılı bir şifrelemenin beklenen seviyelerinde bulunmuştur. Özellikle, NPCR değeri yaklaşık %99,6 ve UACI değeri %50 olarak ölçülmüştür. Bu değerler literatürde belirtilen ideal değerlere yakındır ve algoritmanın güçlü bir avalans etkisine sahip olduğunu göstermektedir . Başka bir deyişle, orijinal girdi veya anahtardaki ufak bir değişiklik, şifreli çıktıda yaygın ve büyük bir farklılığa yol açmaktadır; bu da yöntemin diferansiyel saldırılara karşı dayanıklılığını kanıtlar.

2.4.5 Saldırı senaryoları

Yukarıdaki analizlere ek olarak, geliştirilen şifreleme algoritması anahtar duyarlılığı, kırpma saldırısı ve gürültü saldırısı senaryoları altında da test edilmiştir. Bu testler, algoritmanın pratik koşullarda veya kasıtlı bozma girişimlerinde ne kadar sağlam kaldığını değerlendirmek açısından önemlidir.

2.4.5.1 Anahtar duyarlılığı (Key sensitivity)

Şifreleme algoritmasının anahtardaki küçük değişikliklere ne kadar duyarlı olduğunu ölçmek amacıyla yapılan testtir. Güvenli bir algoritma, sadece 1 bit farkıyla farklılaşan anahtarlar kullanıldığında bambaşka şifreli görüntüler üretmeli ve yanlış anahtarla orijinale ulaşılamamalıdır. Bunu doğrulamak için, aynı düz görüntü iki farklı anahtarla şifrelenmiştir; bu anahtarlar birbirinin sadece bir bitini farklı olacak şekilde seçilmiştir. Sonuç olarak elde edilen iki şifreli görüntü görünür hiçbir benzerlik taşımamaktadır.

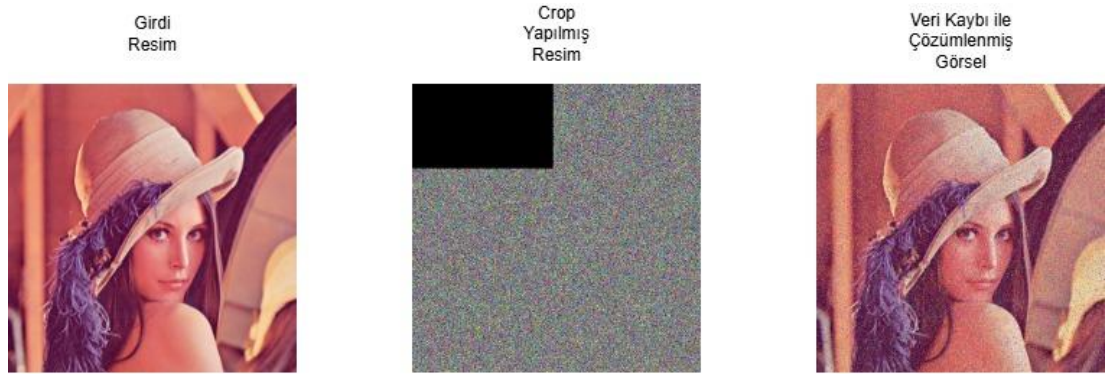


Şekil 2.8: Şifredeki 1 bit değişim sonucu çözümlenen görsel

Şekil 2.8’de bu deneyin çıktıları örneklenmiştir: bir bit değişen anahtarla üretilen şifreli görüntüler arasındaki mutlak fark görüntüsünde neredeyse tüm piksellerin değiştiği görülmektedir ($PSNR = 6.71$ dB). Bu da algoritmanın anahtara karşı son derece hassas olduğunu, bir bitlik anahtar hatasında dahi şifre çözme işleminin başarılı olamadığını gösterir. Nitekim, yanlış (bir bit farklı) anahtarla çözümleme yapıldığında ortaya çıkan görüntü tamamen anlamsız olup orijinal hakkında herhangi bir bilgi içermemektedir. Bu yüksek anahtar duyarlılığı, şifreleme sistemimizin doğru anahtarın kullanılmasını zorunlu kıldığını ve brute-force anahtar denemelerine karşı direnç sağladığını teyit etmektedir.

2.4.5.2 Kırpma saldırısı (Crop attack)

Bu senaryoda, şifreli görüntünün bir bölümü kesilerek (verinin bir kısmı kaybedilerek) algoritmanın hala çözüm üretip üretemeyeceği incelenir. Özellikle, şifreli görüntünün belirli bir kısmı (örneğin kenarından %10-20’lik bir dilim) silinmiş veya iletilmediği varsayılmış, ardından bu eksik veriyle şifre çözme işlemi gerçekleştirilmiştir. Amaç, algoritmanın veri kaybı durumunda dayanıklılığını test etmektir. Literatürde görüntü şifreleme algoritmalarının, şifreli verinin bir kısmı eksilmiş olsa bile orijinal görüntüyü kısmen de olsa doğru şekilde geri getirebilmesi istenen bir özelliktir . Yaptığımız deneylerde, kırılmış (bölgesi eksik) şifreli görüntüler kullanılarak çözümleme yapılmıştır. Sonuç olarak, çıkarılan bölgeye karşılık gelen kısımlar deşifre edilmiş görüntüde kaotiklikten dolayı noise şeklinde dursada, geri kalan kısımların doğru bir şekilde çözülebildiği gözlemlenmiştir.

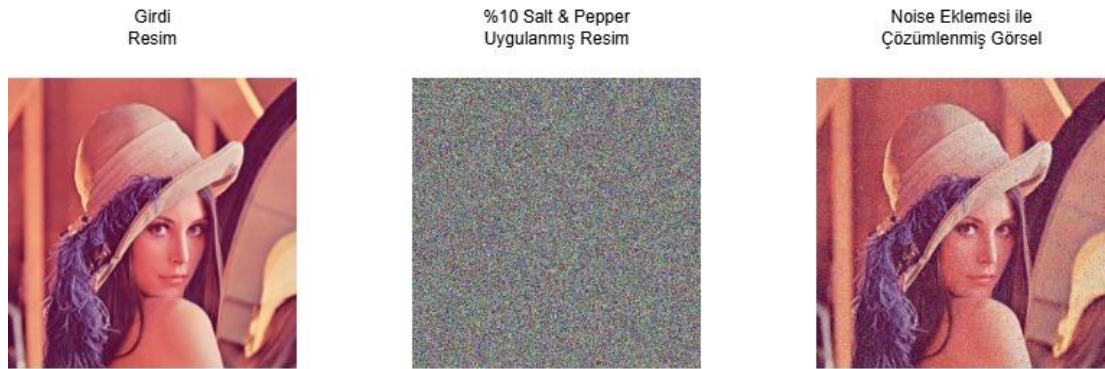


Şekil 2.9: Crop attack uygulanmış verinin çözümlenmesi

Şekil 2.9’da bu duruma ait bir örnek görülmektedir. Bu test, önerilen algoritmanın %12.5’lik veri kaybı (şifreli görüntüde eksilme) durumunda dahi kısmi çözüm üretebildiğini ve tamamen çökmediğini göstermektedir; ölçülen PSNR = 17.93 dB değeri de bu durumu niceliksel olarak doğrulamaktadır. Ancak, kaybolan kısım hakkındaki bilgiler geri alınamadığı için, bu tür bir saldırı orijinal görüntünün o bölgesinde kalıcı bilgi kaybına yol açmaktadır. Yine de, algoritmanın sağlamlığı açısından bakıldığında, kısmen bozulmuş bir şifreli görüntüyle çalışırken bile elde kalan veriden anlamlı içerik çıkarabilmesi önemli bir başarıdır.

2.4.5.3 Gürültü saldırısı (Noise attack)

Bu senaryoda ise iletim sırasında veya kötü niyetli bir girişim sonucu şifreli görüntüye rastgele gürültü eklenmesi durumu ele alınmıştır. Şifreli görüntülere farklı yoğunluklarda Gauss gürültüsü ve tuz-biber (salt-and-pepper) gürültüsü eklenerek, ortaya çıkan bozulmuş şifreli görüntüler çözülmeye çalışılmıştır. Bu test, algoritmanın iletişim gürültüsüne karşı dayanıklılığını ölçmeyi sağlar. Uygulanan düşük seviyeli gürültülerde, şifre çözme sonrasında elde edilen görüntünün büyük oranda orijinali yansıttığı, yalnızca gürültü uygulanan bazı piksel konumlarında hataların (yanlış renk veya parlaklık) olduğu görülmüştür. Şekilde (Şekil 2.10), %10 oranında tuz-biber gürültüsü eklenmiş bir şifreli görüntü çözüldüğünde, orijinal resim genel hatlarıyla doğru bir şekilde ortaya çıkmakta; sadece az sayıda pikselde rastgele siyah/beyaz noktalar gözlenmektedir. Bu senaryoda ölçülen PSNR = 18.39 dB, gürültünün görüntü kalitesini hangi seviyede etkilediğini niceliksel olarak ortaya koymaktadır.



Şekil 2.10: %10 Salt&Pepper uygulanmış verinin çözülmesi

Gürültü yoğunluğu arttıkça, çözülmüş görüntüdeki bozulma da artmakla birlikte görüntünün içeriği belli bir düzeye kadar anlaşılabilir kalmaktadır. Algoritmamızın, şifreli görüntü belirli ölçüde gürültüye maruz kalsa bile işe yarar bir çözüm üretebilmesi, gürültü saldırılarına karşı kısmi bir direnç sağladığını göstermektedir. Özellikle multimedya uygulamalarında, mükemmel hatasızlık yerine içeriğin anlaşılabilir olması öncelikli olabileceğinden, ufak tefek bozulmalarla orijinal içeriğin hala elde edilebilmesi değerli bir özelliktir. Bu açıdan bakıldığında, şifreleme algoritmamız gürültü eklenmiş şifreli görüntülerden orijinal içeriği büyük ölçüde çıkarabilmekte ve böylece olası iletim hatalarına karşı da belirli bir tolerans sunmaktadır.

2.5 Performans Ölçümleri

Bu bölümde geliştirilen görüntü şifreleme algoritmasının zaman ve bellek performans ölçümleri sunulmaktadır. Literatürde benzer kaotik şifreleme yöntemleri genellikle C/C++ gibi düşük seviye dillerde veya MATLAB gibi optimize ortamlar kullanılarak gerçekleştirilmiştir . Buna karşın, bu çalışmada algoritma Python dilinde yazılmış ve kritik hesaplamalar Cython ile hızlandırılmıştır. Bu sayede yüksek seviye bir dil kullanılmasına rağmen makul sürede çalışma sağlanmış; elde edilen süreler literatürde bildirilen bazı düşük seviye uygulamalarla kıyaslanabilir düzeydedir . Aşağıdaki alt bölümlerde, 512×512 ve 1024×1024 boyutlarındaki görüntüler üzerinde yapılan zaman ve bellek ölçümlerinin sonuçları detaylandırılmaktadır.

2.5.1 Zaman ölçümleri

Şifreleme ve şifre çözme işlemleri için gereken süreler, farklı boyutlardaki görüntüler üzerinde ölçülmüştür. Uygulamanın Python ile yazılması ve yalnızca belirli modüllerin Cython ile hızlandırılması nedeniyle, beklenebileceği gibi çalışma süreleri tamamen C/C++ ile geliştirilmiş uygulamalara göre biraz daha yüksek olabilir. Yine de, algoritmanın kaotik ve DNA temelli yapısının getirdiği hesaplama yüküne rağmen elde edilen süreler pratik olarak kabul edilebilir düzeydedir. Örneğin, geleneksel bir şifreleme yöntemi olan AES, optimize edilmiş kütüphaneler sayesinde 512×512 boyutlu bir görüntüyü Python ortamında yalnızca milisaniyeler mertebesinde (yaklaşık 13–19 ms) sürede şifreleyebilmektedir . Kaos tabanlı karmaşık işlemler içeren algoritmamızda ise süreler biraz daha yüksek olmakla birlikte, literatürde benzer yöntemler için bildirilen saniye mertebesindeki sürelerle uyumludur (512×512 görüntüler için ~1,4–6 saniye aralığı gibi) .

Ölçülen ortalama süreler aşağıdaki gibidir:

- 512×512 görüntü: Şifreleme süresi ortalama ~0,8 – 1,1 saniye, şifre çözme süresi ise benzer şekilde ~0,8 – 1,1 saniye aralığında gerçekleşmiştir. Şifreleme ve çözme adımları benzer karmaşıklıkta işlemler içerdiğinden, bu iki işlem için süreler birbirine yakın bulunmuştur.
- 1024×1024 görüntü: Şifreleme süresi ortalama ~4,2 saniye olarak ölçülmüştür. Şifre çözme süresi ise ~3,9 – 4,4 saniye aralığında değişmektedir. Görüntü

boyutu 512×512 'den 1024×1024 'e çıkarıldığında (piksel sayısı 4 katına çıkmaktadır), şifreleme süresinin de yaklaşık 4-5 kat arttığı gözlemlenmiştir. Bu artış, algoritmanın zaman karmaşıklığının görüntüdeki piksel sayısına göre yaklaşık doğrusal ($O(m \cdot n)$) ölçeklendiğini göstermektedir. Gerçekten de benzer kaos-DNA tabanlı şifreleme çalışmalarında 512×512 piksel için 1,4–1,6 s mertebesinde olan şifreleme sürelerinin 1024×1024 için 5–6 s mertebelerine çıktığı literatürde rapor edilmiştir. Dolayısıyla, bizim Python+Cython uygulamamızın elde ettiği ~4 saniyelik süreler, söz konusu yöntemlerle kıyaslandığında oldukça rekabetçidir ve önerilen algoritmanın pratikte zaman açısından uygulanabilir olduğunu teyit etmektedir.

2.5.2 Bellek kullanımı

Algoritmanın bellek kullanımı da farklı boyutlardaki görüntüler için incelenmiştir. Kaotik haritalardan üretilen diziler, DNA kodlamaları ve diğer ara veriler, görüntü boyutuna bağlı olarak oldukça büyük veri yapıları oluşturur. Python dilinin getirdiği ek yük de göz önüne alındığında, uygulamanın bellek gereksinimi düşük seviye bir dille yazılmış eşdeğerine göre daha yüksek olabilmektedir. Yapılan ölçümler, görüntü boyutu arttıkça bellek tüketiminin beklendiği gibi arttığını göstermektedir. Özellikle, algoritmanın her adımında oluşan geçici diziler ve DNA dönüştürmeleri bellek üzerinde ek yük oluşturmaktadır.

Elde edilen bellek kullanım değerleri aşağıda özetlenmiştir:

- 512×512 görüntü: Şifreleme işlemi sırasında ~200–270 MB seviyesinde bellek kullanımına ihtiyaç duyulmuştur (tepe noktası değeri). Bu bellek kullanımına, görüntünün bit dizilerine ve DNA dizilerine dönüştürülmesi, kaotik maskelerin oluşturulması ve bu verilerin aynı anda bellekte tutulması etki etmektedir.
- 1024×1024 görüntü: Girdi boyutu iki katına (piksel sayısı dört katına) çıkarıldığında bellek ihtiyacı da yaklaşık iki kat artarak ~400–480 MB aralığında ölçülmüştür. Bellek artışının, piksel sayısındaki artışa göre lineer olmaması, Python çalışma zamanı ve veri yapılarının sabit ek yüklerinden kaynaklanabilir. Yani, bazı sabit giderler bulunduğundan piksel sayısı dört katına çıksa da toplam bellek kullanımı yaklaşık iki katına yükselmiştir. Buna rağmen, bellek kullanımındaki bu değerler günümüz tipik bir bilgisayar

donanımı için makul düzeydedir ve uygulamanın çalıştığı test ortamında herhangi bir bellek sorununa yol açmamıştır.

Sonuç olarak, Python + Cython ile gerçekleştirilen uygulamada, performans ölçümleri hem süre hem bellek açısından tatmin edici bulunmuştur. Düşük seviye dillerde yazılan uygulamalar kadar optimize olmasa da, önerilen yöntemin 512×512 ve 1024×1024 boyutlu görüntülerde birkaç saniye içinde çalışabilmesi ve makul bellek sınırları içerisinde kalması, yöntemin pratikte uygulanabilir olduğunu göstermektedir. Bu sayede, literatürde düşük seviye dillerle elde edilen sonuçlara yakın performans değerleri, daha yüksek seviye bir geliştirme ortamında elde edilmiştir. Geliştirilen Cython modülleri (dna_codec_cy , logistic_cy , chaos_utils_cy gibi) özellikle DNA kodlama/çözme, kaotik dizi üretimi ve XOR işlemleri gibi yoğun hesaplama gerektiren bölümlerde önemli hız kazanımları sağlamıştır. İlerleyen çalışmalar kapsamında, algoritmanın daha verimli bellek yönetimi ve daha hızlı kaotik sayı üretimi gibi yönlerde optimize edilmesiyle, performansın düşük seviye dil uygulamalarına daha da yaklaştırılması mümkün görülmektedir.

2.6 Web Arayüzü ve API Entegrasyonu

Bu çalışma, geliştirilen DNA tabanlı şifreleme algoritmasını yalnızca teorik olarak değil, aynı zamanda pratik olarak da uygulamaya olanak sağlamak amacıyla bir web arayüzü ve RESTful API servisi ile desteklemektedir. Bu sayede kullanıcılar, herhangi bir programlama bilgisi olmadan bir görseli yükleyip şifreleyebilmekte, istedikleri zaman bu şifreli veriyi çözebilmektedir. Sistem ayrıca analiz, test ve parametre ayarlama işlevlerini de sunar.

2.6.1 Kullanılan teknolojiler

- **Backend:** Projenin sunucu tarafı Python dili ile geliştirilmiş olup Flask mikro çatısı kullanılmıştır. CORS desteği sağlanarak istemci tarafından gelen çapraz isteklerin yönetimi sağlanmıştır.
- **Frontend:** Kullanıcı arayüzü Bootstrap 5 ve jQuery desteğiyle oluşturulmuştur. Sade, responsive ve interaktif bir tasarım tercih edilmiştir.

2.6.2 Temel uç noktalar (Endpoints)

Sistem, şu REST API endpoint'leri üzerinden işlem yapar:

- **/encrypt [POST]:** Şifreleme için kullanılır. Gönderilen görsel dosyası, parola ve opsiyonel DNA parametrelerine göre şifrelenir.

Girdi: image (görsel dosyası), key (parola), dna_rule, x0[], r[]

Çıktı: JSON içinde şunlar dönür: image_url, directory, duration

- **/decrypt [POST]:** Şifreli PNG dosyasını çözer.

Girdi: image (veya directory), key

Çıktı: JSON içinde şunlar dönür: image_url, directory, duration

- **/analyze [POST]:** Histogram, korelasyon, PSNR, NPCR ve UACI analizlerini yapar.

Girdi: input ve output adlı iki dosya

Çıktı: hist_url, corr_url, psnr, npcr, uaci

- **/crop_attack [POST]:** Seçilen alanı siyah maskeyle kaplayarak sınırlı veriyle çözme testi yapar.

- **/noise_attack [POST]:** Gaussian veya Salt-Pepper şeklinde gürültü ekleyip çözme performansını test eder.
- **/example_images [GET]:** Hazır görsel örneklerini istemciye döner.

2.6.3 Arayüzden iş akışı

HTML tabanlı arayüz, kullanıcı dostu bir deneyim sunmak üzere sürükle-bırak desteği, yükleme butonları ve modal pencerelerle desteklenmiştir. Kullanıcı şu adımlarla işlem gerçekleştirir:

1. **Görsel Yükleme:** Sürükleyerek veya "Yükle" butonuyla bir görsel seçilir.
2. **Parola ve Parametre Girişi:** Kullanıcı bir parola belirler ve isterse DNA şifreleme parametrelerini (dna_rule, x0, r) değiştirir.
3. **Şifreleme:** "Şifrele" butonuna tıklandığında /encrypt endpoint'ine istek atılır ve çıktı görseli ekranda belirir.
4. **Çözme:** Önceden şifrelenmiş bir PNG tekrar yüklenerek ve parola girilerek, "Çöz" butonuyla /decrypt endpoint'i çağrılır.
5. **Analiz & Testler:** Histogram, korelasyon, PSNR, NPCR ve UACI analizleri /analyze endpoint'iyle hesaplanır ve sonuçlar arayüzde görsel olarak sunulur. Ayrıca crop, noise ve key sensitivity testleri modal formlar ile kolayca uygulanabilir.

Bu bileşenler, algoritmanın yalnızca akademik değil, aynı zamanda kullanılabilir bir prototip çözüme dönüşmesini sağlamıştır.

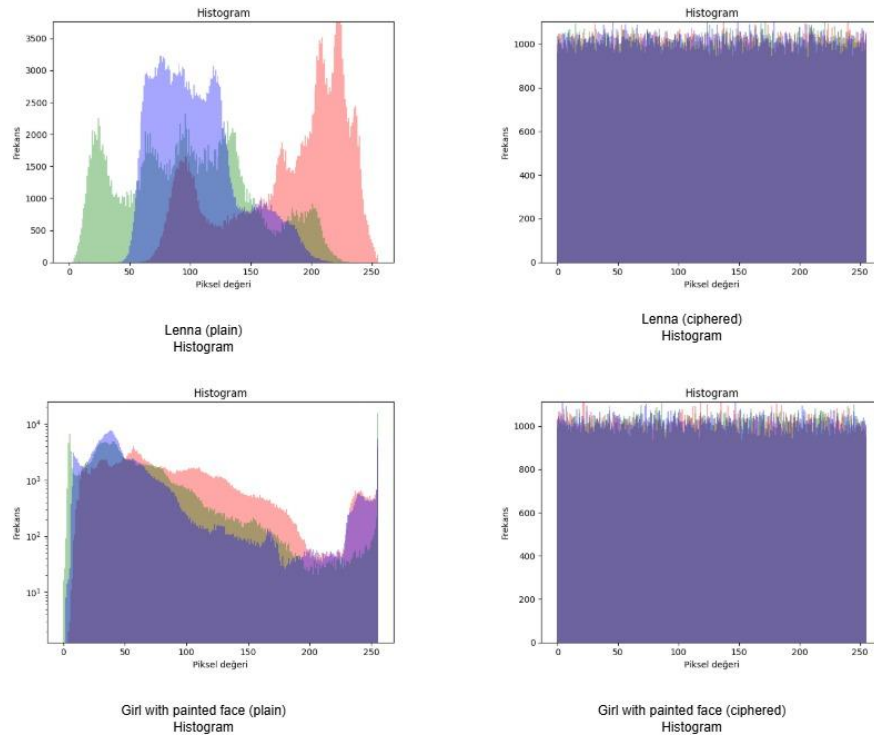
3. SONUÇLAR VE DEĞERLENDİRME

3.1 Şifreleme Kalitesi Sonuçları

Bu bölümde, geliştirilen DNA tabanlı görüntü şifreleme algoritmasının temel şifreleme kalitesi metrikleri açısından performansı sunulmaktadır. Testler, literatürde yaygın olarak kullanılan beş farklı standart test görseli üzerinden gerçekleştirilmiştir: **Lenna**, **Pepper**, **Cameraman**, **Baboon** ve **Boat**. Her bir görüntü için orijinal (plain) ve şifreli (ciphered) halleri üzerinden **histogram dağılımı**, **komşu piksel korelasyon katsayıları** ve **bilgi entropisi** metrikleri analiz edilmiştir.

3.1.1 Histogram analizi

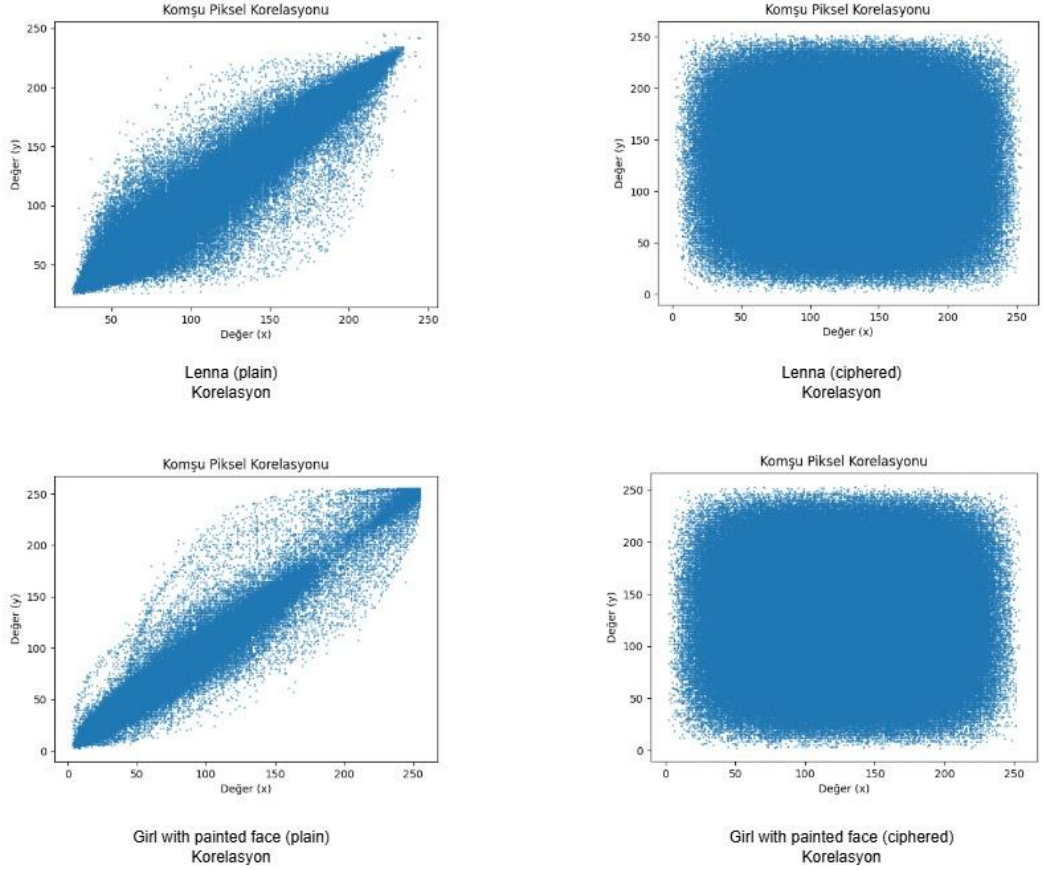
Histogram analizi, şifreli görüntünün istatistiksel dağılımını gözlemlemek için en temel görsel ölçüm aracıdır. Orijinal görüntülerde, R-G-B kanallarında genellikle belirgin tepe noktaları ve yapısal yoğunluklar görülürken, şifreli görüntülerin histogramlarının büyük oranda düzleştiği gözlemlenmiştir. Bu, görüntünün istatistiksel özelliklerinin bozulduğunu ve sıradan analiz teknikleriyle anlaşılabilirliğinin ortadan kalktığını göstermektedir. Orijinal ve şifreli hallerin histogramları Şekil 3.1’de örneklenmiştir:



Şekil 3.1: Lenna ve girl plain ve ciphered histogram farkları

3.1.2 Komşu piksel korelasyonu

Bir görüntüdeki komşu pikseller arasındaki korelasyon, şifrelemenin etkinliğini ortaya koyan en önemli istatistiksel testlerden biridir. Orijinal görüntülerde, yatay, dikey ve çapraz yönlerde çok yüksek korelasyon (genellikle 0.9'un üzerinde) görülürken, şifreli görüntülerde bu değerler çarpıcı bir şekilde düşüş göstermiştir (bkz. Şekil 3.2). Yapılan testlerde, şifreli görüntüler için ortalama korelasyon değerleri **-0.0009** ila **-0.00013** arasında bulunmuş, bu da verideki yapısal ilişkilerin tamamen ortadan kaldırıldığını göstermiştir. Korelasyonlar scatter plot şeklinde sunulmuş ve şu şekillerle desteklenmiştir:



Şekil 3.2: Entropi scatterplot'u – Lenna/Girl (orijinal-şifreli)

3.1.3 Bilgi entropisi

Entropi, Shannon’ın bilgi kuramındaki tanımına göre bir kaynağın ürettiği sembollerin belirsizlik düzeyini ölçer ve 8-bit renk uzayında teorik olarak en yüksek 8,0 bit/piksel değerini alabilir. Bu çalışmada kullanılan beş test görselinin şifrelenmeden önceki entropi değerleri 6,95 bit/piksel (Girl) ile 7,27 bit/piksel (Lenna) aralığında değişirken, şifreleme sonrasında tüm görüntüler için entropinin 7,9992–7,9998 bit/piksel bandına yükseldiği saptanmıştır (Çizelge 2.2). Başka bir ifadeyle, ortalama entropi artışı yaklaşık 0,9–1,2 bit olup, her bir görüntü neredeyse teorik üst sınıra erişecek ölçüde rastgele hâle gelmiştir. İdeal değerle (8,0) arasındaki farkın en çok $7,7 \times 10^{-4}$ bit gibi ihmal edilebilir düzeyde kalması, histogramın pratik olarak tam bir uniform dağılım sergilediğini; dolayısıyla sembol olasılıklarının eşitlenerek istatistiksel öngörü yapılmasını zorlaştırdığını göstermektedir. Literatürde 7,5–7,99 bit/piksel aralığı “çok iyi” olarak kabul edildiğinden, elde edilen 7,999 bit/piksel civarındaki değerler algoritmanın rastgelelik üretme kapasitesini üst sınırdaki karşıladığını kanıtlar niteliktedir. Bu yüksek entropi aynı zamanda, histogram tabanlı veya düşük seviyeli istatistiksel kriptanaliz yöntemlerine karşı koruma sağlar; çünkü şifreli görüntü, görsel açıdan beyaz gürültüye oldukça yakın bir nitelik kazanır. Dahası, anahtar hassasiyeti testlerinde kaydedilen yüksek NPCR ve UACI oranları ile birlikte değerlendirildiğinde, algoritmanın yalnızca tek bitlik anahtar değişimlerine dahi kuvvetli tepki verirken, çıktı dağılımını istatistiksel olarak da karmaşıktırdığı anlaşılmaktadır.

Çizelge 2.2 : Plain ve ciphred görsellerin entropileri.

	Lenna	Girl	Sail boats	Baboon	Stonebuilding
Plain	7,27186	6,95068	7,24404	6,81145	7,09933
Ciphred	7,99929	7,99923	7,99982	7,99937	7,99948

3.2 Diferansiyel Dayanıklılık Analizi

Bir şifreleme algoritmasının en kritik güvenlik kriterlerinden biri, diferansiyel analizlere karşı olan direncidir. Bu analiz, görüntünün sadece bir pikselinde yapılan ufak bir değişikliğin, şifreli çıktıda ne kadar yayıldığını ve bu değişikliğin sonucunda algoritmanın yeterince "karışıklık" (confusion) sağlayıp sağlamadığını gösterir. Bu bağlamda en çok kullanılan iki metrik: **NPCR (Number of Pixels Change Rate)** ve **UACI (Unified Average Changing Intensity)** değerleridir.

Bu çalışmada, şifrelenmiş bir görüntü ile orijinali arasındaki fark temelinde NPCR ve UACI analizleri uygulanmıştır. Analizler, beş farklı test görseli üzerinde tekil şifreleme sonucuna dayalı olarak gerçekleştirilmiştir. Sonuçlar Çizelge 2.3'te sunulmuş olup, tüm görüntüler için NPCR değerlerinin %99.60'ın üzerinde, UACI değerlerinin ise yaklaşık %31 civarında olduğu görülmektedir.

Bu durum, algoritmanın görüntü üzerindeki minimal değişiklikleri şifreli yapıya etkili şekilde yayabildiğini ve "**avalanche effect**" ilkesiyle tutarlı davrandığını göstermektedir.

Çizelge 2.3: Şifrelenmiş görsellerin NPCR ve UACI değerleri.

	Lenna	Girl	Sailboats	Baboon	Stonebuilding
NPCR (%)	99,6090	99,6056	99,6136	99,6096	99,6108
UACI (%)	30,4182	35,4568	28,0312	31,0480	28,3386

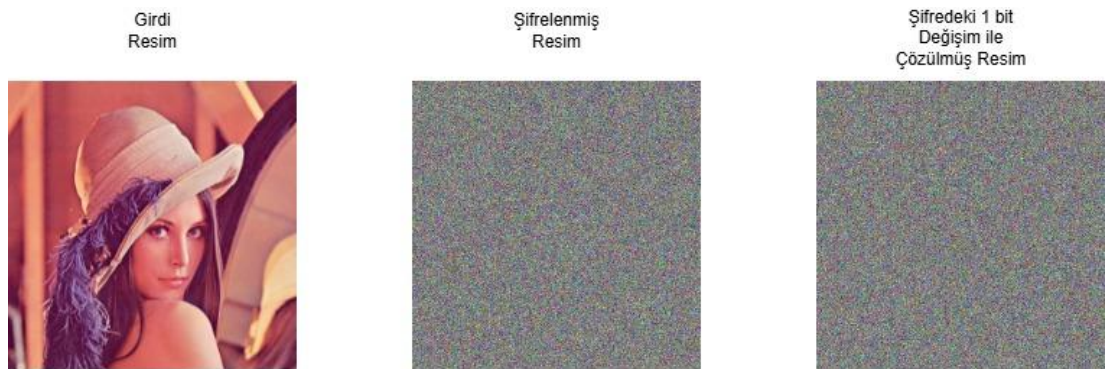
Ortalama olarak NPCR değerleri %99,61, UACI ise %31,65 civarında hesaplanmıştır. Bu değerler, literatürde genellikle kabul edilen %99,3 ve %30,0 eşiklerinin üzerinde olup, önerilen algoritmanın diferansiyel dayanıklılığını tatmin edici düzeyde sağladığını göstermektedir. Çalışma tekrar sayısı sınırlı olsa da elde edilen bulgular, algoritmanın istikrarlı bir şekilde farklı görsel girdilere karşı benzer dayanıklılık sağladığını ortaya koymaktadır.

3.3 Saldırı Dayanımı Deneyleri

Kriptografik bir algoritmanın güvenlik değerlendirmesinde klasik istatistiksel metriklerin yanı sıra, algoritmanın pratik siber sınıflandırmalara ve çeşitli veri bozulmalarına karşı nasıl tepki verdiği de önemli bir göstergedir. Bu kapsamda, önerilen DNA tabanlı şifreleme sisteminin şu tür saldırı senaryoları altında dayanıklılığı test edilmiştir: Anahtara duyarlılık (Key Sensitivity), Kırpma saldırısı (Crop Attack), ve Gürültü saldırısı (Noise Attack).

3.3.1 Anahtara duyarlılık (Key sensitivity)

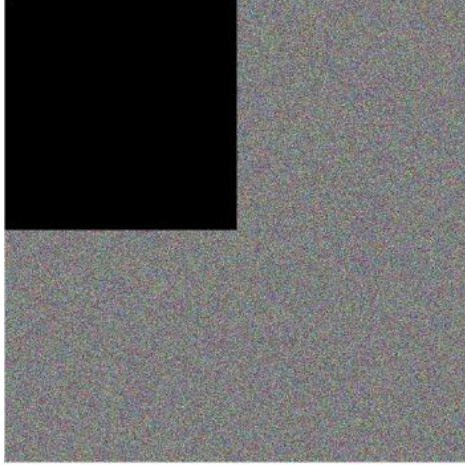
Parola tabanlı sistemlerde, ideal durumda, şifreleme anahtarında yapılacak tek bitlik bir değişiklik bile çözüm çıktısını tamamen anlamsız hâle getirmelidir. Yapılan testlerde, örneğin “ilhan123” şifresiyle şifrelenmiş bir görüntü, şifre bit düzeyinde manipüle edilerek yalnızca ilk 1 bit değiştirildiğinde (“élhan123”) çözüldüğünde, çıktı görüntünün farklılığı hemen gözlemlenmiştir. Orijinal görüntüye kıyasla çıktı tamamen rastgele dağılmış gürültü şeklinde olup veriye geri dönüşü anlamsız kılmaktadır. Yanlış şifreyle çözümlenen bu görüntü için ölçülen metrikler PSNR = 8,614 dB, NPCR = 99,6002 % ve UACI = 30,4539 % olarak elde edilmiştir. Düşük PSNR ile birlikte yüksek NPCR ve UACI değerleri, algoritmanın anahtar hassasiyeti açısından beklenen sert tepkiyi verdiğini doğrulamaktadır. Bu durum, Şekil 3.3’te yanlış şifreyle çözümlenen görselde de açıkça görülmektedir.



Şekil 3.3 : Şifredeki 1 bit değişim sonucu çözümlenen görsel

3.3.2 Kırpma saldırısı (Crop attack)

Görüntünün belirli bir bölgesinin fiziksel olarak kaybedilmesinin (kırılmasının) şifre çözüm sürecindeki etkisini incelemek amacıyla, arayüzde yer alan “crop-attack” modülü kullanılarak Sail Boats görselinin şifreli PNG biçimindeki sürümünün tam %25’i kesilerek bilinçli veri kaybı oluşturulmuştur (bkz. Şekil 3.4). Otomatik kırpma modülü, seçilen bölgede pikselleri kalıcı olarak sıfırlamış ve bu işlem sırasında çerçeve parametreleri de kayıt altına alınmıştır. Kırılmış veriye rağmen yapılan çözüm denemesinde, eksik bölgelere karşılık gelen alanlarda belirgin bozulmalar görülmüş; buna karşın kaybolmayan kısımlarda nesne konturları ve renk düzenleri okunabilir düzeyde kalmıştır. Sayısal değerlendirme olarak hesaplanan PSNR değeri 14.3448 dB olup, görsel kalitenin orta düzeyde korunabildiğini, ancak bölgesel kayıpların algılanabilir düzeyde bozulmaya yol açtığını göstermektedir. Bu sonuç, önerilen algoritmanın bölgesel veri kaybı senaryolarında sınırlı bir tolerans sağladığını, ancak tam görsel bütünlüğü korumak için ileri düzey kayıp telafisi stratejilerine ihtiyaç duyulabileceğini ortaya koymaktadır.



%25 Crop Yapılmış
Sailboats Ciphred

PSNR : 14.3448

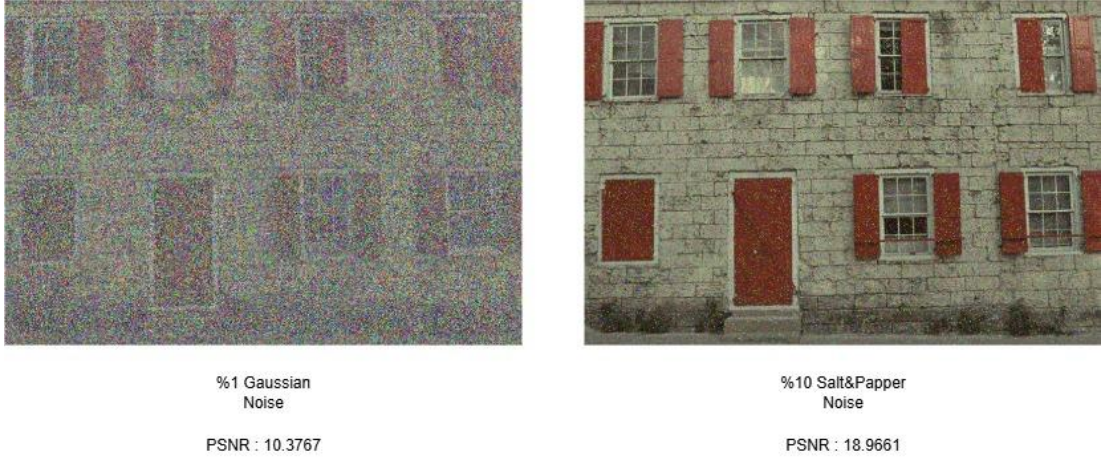


Çözümlemiş
Görsel

Şekil 3.4: Kırılmış şifreli görselin çözülmüş hali (Sail boats görseli)

3.3.3 Gürültü saldırısı (Noise attack)

Görsel verilerde rastgele bozulmalar yaratan gürültü türleri, algoritmanın esnekliğini sınamak için yaygın olarak kullanılmaktadır. Bu çalışmada, arayüzde yer alan “noise-attack” modülü aracılığıyla şifrelenmiş görüntüler üzerine %1 oranında Gaussian gürültü ve %10 oranında Salt & Pepper gürültüsü enjekte edilmiş, ardından bu bozulmuş görüntüler çözülerek çıktılar incelenmiştir. Gürültü eklenmiş halleri değerlendirilirken, genel yapıların hâlen tanınabilir kaldığı ve sistemin belirli bir hata toleransı sergilediği gözlemlenmiştir; ancak özellikle Gaussian gürültü altında elde edilen 10.3767 dB’lik PSNR değeri, bu tür bozulmalara karşı dayanımın iyileştirilebileceğine işaret etmektedir.



Şekil 3.5: Gaussian ve salt & pepper gürültü uygulanmış görsellerin çözülümü

Şekil 3.5’teki bulgular, önerilen algoritmanın rastgele (Gaussian) ve impulsif (Salt & Pepper) gürültü türlerine karşı belirli bir hata toleransı sergilediğini, ancak Gaussian gürültü altında elde edilen 10,3767 dB’lik PSNR değerinin iyileştirme gerektirdiğini göstermektedir. Bu nedenle, gelecekteki çalışmalarda gürültü dayanımının özellikle Gaussian bozulmalar karşısında artırılmasına odaklanılması önerilmektedir.

3.4 Genel Değerlendirme

Önerilen DNA tabanlı görüntü şifreleme yöntemi, hem teorik hem de deneysel boyutta bir dizi analizden geçirilerek güvenlik ve performans açısından değerlendirilmiştir. Bu bölümde, önceki alt bölümlerde sunulan bulgular ışığında algoritmanın genel yeterliliği yorumlanmaktadır.

Histogram analizi ve entropi hesapları, algoritmanın şifreli veride yüksek seviyede rastgelelik sağladığını ortaya koymuş; pikseller arası korelasyon katsayıları ise orijinal görselde %90'ın üzerindeyken, şifreli versiyonda çok düşük değerlere (yaklaşık 0,001 altına) inerek algoritmanın istatistiksel saldırılara karşı dirençli olduğunu göstermiştir.

Diferansiyel dayanıklılık açısından, NPCR değerleri tüm testlerde %99,6'nın üzerinde, UACI ise %31,65 civarında hesaplanmış olup, bu oranlar algoritmanın "avalanche etkisi" göstermede yeterli seviyede olduğunu kanıtlamıştır.

Gerçek hayata dönük sınırlı bozulma senaryoları (anahtar hassasiyeti, kırpma ve gürültü saldırıları) altında yapılan testler, algoritmanın anahtar hassasiyeti açısından beklenenden daha yüksek bir sertlik sergilediğini, ancak Gaussian gürültü altındaki performansının görece zayıf kaldığını göstermiştir. Şifredeki tek bitlik bir fark, çoğu senaryoda beklenen “hafif gürültü benzeri” etkinin ötesinde belirgin bozulmalara yol açarken, %1’lik Gaussian gürültü uygulamasında PSNR değerinin 10.3767 dB’ye düşmesi gürültü dayanımının geliştirilmesi gerektiğini ortaya koymaktadır. Bu durum, gelecekteki çalışmalarda gürültü saldırılarına karşı direnç artırımı üzerine odaklanılması gerektiğini işaret etmektedir.

Performans açısından değerlendirme yapıldığında ise, Python dili ile geliştirilmiş olan uygulamanın Cython modül desteği sayesinde 512×512 boyutunda bir görseli şifrelemede ortalama 1 saniye gibi kabul edilebilir bir sürede çalıştığı; 1024×1024 görsellerde ise bu sürenin yaklaşık 4 saniyeyi bulduğu gözlenmiştir. Bellek kullanımı ise modern sistemler için makul sınırlar içerisinde kalarak 200–400 MB aralığında seyretmektedir.

Genel olarak, bu tez kapsamında geliştirilen DNA tabanlı görüntü şifreleme yöntemi, hem istatistiksel hem diferansiyel hem de sümülatif sınıf testlerinde tatmin edici seviyelerde performans sergilemiş ve sade, şifrelenmiş dosya formatlarıyla uyumlu bir çözüm sunmuştur.

4. ÖNERİLER

Bu çalışma kapsamında, DNA tabanlı kodlama teknikleri ile kaotik sistemlerin birleştirildiği, parola temelli bir görüntü şifreleme algoritması tasarlanmış ve hem teorik açıdan hem de deneysel metriklerle değerlendirilmiştir. Yüzlerce yıldır temel alanlarda çalışan kriptografik yöntemlerin aksine, DNA benzeri biyolojik temsillerin dijital sistemlere entegre edilmesi, verinin daha çok katmanlı ve anlamlı şekilde gizlenmesini sağlamaktadır. Önerilen sistem, renkli görüntüleri tek boyutlu bayt dizisine dönüştürerek bu dizi üzerinde hem DNA kodlaması hem de kaotik XOR, mask, permütasyon gibi çok aşalı sistematik işlemler uygulamakta, şifreli çıktıyı ise PNG formatında gömme metadata ile birlikte sunmaktadır.

Yapılan testlerde, şifreleme sürecinin istatistiksel anlamda başarılı olduğu; histogram, entropi, korelasyon gibi metriklerde klasik yaklaşımlarla uyumlu ya da daha üst performans gösterdiği gözlemlenmiştir. NPCR ve UACI gibi diferansiyel analiz testlerinde ise %99,6 üzeri NPCR ve % 31,65 bandında UACI değerleri, önerilen algoritmanın "avalanche effect" (sel etkisi) gibi kritik kriptografik ilkeleri yerine getirdiğini ortaya koymuştur. Performans testleri, Python gibi yorumlamalı bir dil kullanılmasına rağmen, Cython modül desteği sayesinde ortalama 1-4 saniye aralığında şifreleme/ çözme süreleri sunmuş, 200-480 MB arasında değişen bellek tüketimi ile modern sistemlerde uygulanabilirliğini göstermiştir.

Ancak çalışmanın belirli sınırlılıkları da dikkate alınmalıdır. Şifre hassasiyetinin bazı durumlarda tam anlamıyla rastgele bozunma yaratmadığı, özellikle benzer şifrelerin kullanımında görsel benzerliğin hala kısmen korunabildiği fark edilmiştir. Bununla birlikte, algoritma bölgesel veri kayıpları (crop attack) ve çok düşük seviyeli noise saldırıları altında da çözülebilir görüntüler üretebilmiştir.

Bu bulgular doğrultusunda, çalışmanın ileriye yönelik geliştirilebileceği alanlar şu şekilde sıralanabilir:

- Anahtara duyarlılık karakteristiği daha sert hale getirilerek, benzer şifrelerle bile anlamlı geri dönüşlerin önü kesilmelidir.
- DNA kodlama kuralı ("dna_rule") sabit değil, görüntünün istatistiksel özelliklerine göre adaptif seçilebilecek bir yapı haline getirilmelidir.
- Şifreleme sistemine ilave olarak, çözülen verinin bütünlüğünü sağlayacak SHA256 tabanlı bir hash veya dijital imza mekanizması eklenebilir.

- Gerçek zamanlı sistemlere uygulanabilirliğini artırmak adına, C++ veya Rust gibi düşük seviyeli dillere taşınması ve gömülü sistemlere port edilmesi faydalı olabilir.
- Web arayüzü üzerinden şifreleme/ çözme hizmeti sunulmasına devam edilirken, istemci tarafında şifreleme yapılması ve sadece şifreli verinin sunucuya gönderilmesi, gizlilik açısından daha üst seviyede bir yaklaşım sağlayabilir.

Sonuç olarak, bu çalışma bir yandan DNA şifreleme tabanlı akademik yaklaşımı gerçek uygulama ortamlarına uyarlamaya çalışırken, diğer yandan sade bir kod yapısı ve web arayüz üzerinden kullanıcı dostu bir deneyim sunmayı hedeflemiştir. Ortaya çıkan sistem, gelecekte daha da geliştirilerek hem akademik hem de pratik açıdan yaygınlaşabilir bir alternatif sunma potansiyeline sahiptir.

KAYNAKLAR

- [1] **B. Schneier**, **Applied Cryptography: Protocols, Algorithms, and Source Code in C**, John Wiley & Sons, 1996.
- [2] **J. Fridrich**, “Image encryption based on chaotic maps,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 9, pp. 928–932, 2001.
- [3] **H. Patidar, K.K. Sud, N. Pareek**, “A new substitution–diffusion-based image cipher using chaotic standard and logistic maps,” *Commun. Nonlinear Sci. Numer. Simul.*, vol. 14, no. 7, pp. 3056–3075, 2009.
- [4] **M. Solak, M. Aras, M. Isik**, “A secure and robust chaotic-based image encryption algorithm using bit-level permutation,” *Nonlinear Dyn.*, vol. 70, pp. 1377–1386, 2012.
- [5] **J.-G. Pak, E.-W. Huang**, “An efficient image encryption algorithm based on one-dimensional chaotic maps,” *Opt. Lett.*, vol. 40, no. 23, pp. 5464–5467, 2015.
- [6] **A. Amina, M. El-Bhiri**, “Image encryption based on improved chaotic map and SHA-256 hash function,” *Signal Process.*, vol. 130, pp. 417–426, 2017.
- [7] **H. Zhang, X. Zeng, G. Guan**, “Image encryption based on five-dimensional conservative hyperchaotic map,” *Nonlinear Dyn.*, vol. 81, pp. 1871–1889, 2015.
- [8] **X. Cao, Y. Xu, W. Zhu**, “A novel hyperchaotic map and its application in image encryption,” *IEEE Access*, vol. 6, pp. 19765–19778, 2018.
- [9] **E. Yavuz, R. Yazıcı, M.C. Kasapbaşı, E. Yamaç**, “A chaos-based image encryption algorithm with simple logical functions,” *Comput. Electr. Eng.*, vol. 54, pp. 471–483, 2016.
- [10] **H. Çavuşoğlu, M. Marouf, H. Ulutaş**, “A hybrid RC6-based S-AES image cipher with improved S-Box design,” *Secur. Commun. Netw.*, vol. 2018, Article ID 3965231, 2018.
- [11] **E. Yavuz**, “A novel chaotic image encryption algorithm based on content-sensitive dynamic function switching scheme,” *Opt. Laser Technol.*, vol. 114, pp. 224–239, 2019.
- [12] **National Institute of Standards and Technology**, **A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications**, NIST SP 800-22 Rev.1a, 2010.

ÖZGEÇMİŞ

Ad-Soyad : İlhan Emre ADAK
Doğum Tarihi ve Yeri : 17.07.2002 – Balıkesir
E-posta : adak.ie@hotmail.com

BİTİRME ÇALIŞMASINDAN TÜRETİLEN MAKALE, BİLDİRİ VEYA SUNUMLAR:

- **Görüntü Verileri İçin DNA Tabanlı Şifreleme Algoritması– Bitirme Sunumu**