T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

KRİPTOGRAFİ-BİLGİ GÜVENLİĞİ İÇİN RASTGELE SAYI ÜRETECİ GELİŞTİRİLMESİ

Taha ETEM

Doktora Tezi

ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

Telekomünikasyon Bilim Dalı

KASIM 2022

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Elektrik Elektronik Mühendisliği Anabilim Dalı

Doktora Tezi

KRİPTOGRAFİ-BİLGİ GÜVENLİĞİ İÇİN RASTGELE SAYI ÜRETECİ GELİŞTİRİLMESİ

Tez Yazarı
Taha ETEM

Danışman
Doç. Dr Turgay KAYA

KASIM 2022
ELAZIĞ

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Elektrik Elektronik Mühendisliği Anabilim Dalı

Doktora Tezi

Başlığı:	Kriptografi-Bilgi Güvenliği için Rastgele Sayı Üreteci Geliştirilmesi
Yazarı:	Taha ETEM
İlk Teslim Tarihi:	14.10.2022
Savunma Tarihi:	15.11.2022

TEZ ONAYI

Fırat Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına göre hazırlanan bu tez aşağıda imzaları bulunan jüri üyeleri tarafından değerlendirilmiş ve akademik dinleyicilere açık yapılan savunma sonucunda OYBİRLİĞİ ile kabul edilmiştir.

		İmza
Danışman:	Doç. Dr Turgay KAYA Fırat Üniversitesi, Mühendislik Fakültesi	Onayladım
Başkan:	Prof. Dr. Yakup DEMİR Fırat Üniversitesi, Mühendislik Fakültesi	Onayladım
Üye:	Prof. Dr. Ali Emre PUSANE Boğaziçi Üniversitesi, Mühendislik Fakültesi	Onayladım
Üye:	Prof. Dr. Taner TUNCER Fırat Üniversitesi, Mühendislik Fakültesi	Onayladım
Üye:	Prof. Dr. Melih Cevdet İNCE Malatya Turgut Özal Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi	Onayladım

Bu tez, Enstitü Yönetim Kurulunun ...../...../20..... tarihli toplantısında tescillenmiştir.

İmza

Prof. Dr. Burhan ERGEN
Enstitü Müdürü

BEYAN

Fırat Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım "Kriptografi-Bilgi Güvenliği için Rastgele Sayı Üreteci Geliştirilmesi" Başlıklı Doktora Tezimin içindeki bütün bilgilerin doğru olduğunu, bilgilerin üretilmesi ve sunulmasında bilimsel etik kurallarına uygun davrandığımı, kullandığım bütün kaynakları atıf yaparak belirttiğimi, maddi ve manevi desteği olan tüm kurum/kuruluş ve kişileri belirttiğimi, burada sunduğum veri ve bilgileri unvan almak amacıyla daha önce hiçbir şekilde kullanmadığımı beyan ederim.

15.11.2022

Taha ETEM

ÖNSÖZ

Teknoloji çağında yaşamanın bir sonucu olarak her geçen gün bilgi güvenliği konusu daha önemli hale gelmektedir. Rastgele sayı üreteçlerinin bu alandaki kullanımı da gün geçtikçe artmaktadır. Bilgi güvenliği alanındaki ihtiyaca binaen başladığım tez çalışmasında birçok yeni çalışma konusu olduğunu fark ettim. Tez konumuz çerçevesinde yaptığımız çalışmalar gösterdi ki kriptoloji biliminin birçok yeni araştırmaya ihtiyacı var. Günümüzde popülaritesi çok yüksek olan bu konu çalışmalar esnasında en büyük motivasyonum oldu.

Tez çalışmamda öncelikle bana en büyük bilimsel katkıyı sunan tez danışmanım Doç. Dr. Turgay KAYA'ya, iş yerimde bana kolaylık sağlayıp tez çalışmalarımı bitirmem de yardımcı olan Doç. Dr. Hüseyin KOÇ'a, çalışmalarım esnasında bana sabırla yardımcı olan, her türlü kolaylığı sağlayan eşim Gamze ETEM'e, bize olan desteğini sürekli hissettiğim annem Fatma ETEM ve babam Esat ETEM'e teşekkürlerimi iletmeyi bir borç bilirim.

Taha ETEM
ELAZIĞ, 2022

İÇİNDEKİLER

Sayfa

ÖZET

Kriptografi-Bilgi Güvenliği için Rastgele Sayı Üreteci Geliştirilmesi

Taha ETEM

Doktora Tezi

FIRAT ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü

Elektrik Elektronik Mühendisliği Anabilim Dalı

Kasım 2022, Sayfa: xiv + 111

Kriptolojik amaçlarla geliştirilen sistemler, gün geçtikçe önemini arttırmaktadır. Rastgele sayı üreteçleri ise kriptoloji biliminin temelinden itibaren süre gelen geniş bir çalışma alanıdır. Bu tez çalışmasında, kriptolojik amaçlarla kullanılmak üzere hem gerçek hem de sözde rastgele sayı üreteci tasarımları gerçekleştirilerek çeşitli testler yardımıyla başarıları ölçülmeye çalışılmıştır.

Tez çalışmasında ilk olarak farklı rastgele sayı üreteci tasarımları gerçekleştirilmiştir. Fiziksel bir sinyale ihtiyaç duyan gerçek rastgele sayı üreteçleri, algoritmik yapıdaki sözde rastgele sayı üreteçleri ile desteklenmiştir. Fiziksel sinyaller, Lyapunov üstelleri ve Hoshen-Kopelman algoritmasıyla ön testten geçirilmiştir. Daha sonra geliştirilen sistemler NIST istatistiksel rastgelelik testleri, otokorelasyon testi, scale-index testi gibi analizlerle doğrulanmaya çalışılmıştır. Burada elde edilen rastgele bit dizileri, mevcut şifreleme algoritmaları ve kendi geliştirdiğimiz şifreleme algoritmasının yardımıyla şifreleme işlemleri için kullanılmıştır. Şifreleme işlemleri için histogram analizi, anahtar hassasiyeti ve boyutu analizi, NPCR ve UACI testi, entropi testi, korelasyon katsayılarının hesaplanması gibi farklı kriptanaliz yöntemleri uygulanarak sistemlerin güvenilirlikleri incelenmiştir.

Bu çalışmada güvenlik gereksinimlerini sağlayan farklı gerçek ve sözde rastgele sayı üreteci tasarımlarıyla yeni bir simetrik blok şifreleme algoritması tasarlanmıştır. Geliştirdiğimiz rastgele sayı üreteçleri istatistiksel testlerden başarılı sonuçlar elde ederek kriptolojik uygulamalarda kullanılabileceği gösterilmiştir. Geliştirilen simetrik blok şifreleme algoritması ise güvenlik testlerinden başarı sağlamasının yanı sıra, şifreleme ve şifre çözme işlemlerinde standart algortimalardan daha hızlı çalıştığı yaptığımız çalışmada gösterilmiştir.

Anahtar Kelimeler: Rastgele sayı üreteci, Kriptoloji, Şifreleme, NIST istatistiksel testleri

ABSTRACT

Random Number Generator Design for Cryptography-Information Security

Taha ETEM

Ph.D. Thesis

FIRAT UNIVERSITY
Graduate School of Natural and Applied Sciences

Department of Electrical and Electronics Engineering

November 2022, Pages: xiv + 111

Systems developed for cryptological purposes are increasing their importance day by day. On the other hand, random number generators have a wide field of study that has been going on since the existence of cryptology. In this thesis, both real and pseudo-random number generator designs to be used for cryptological purposes were realized and their success was tried to be measured with the help of various tests.

In the beginning of the thesis study, different random number generator designs were carried out. True random number generators that need a physical signal are supported by pseudo random number generators in algorithmic structure. Physical signals are pretested with Lyapunov exponents and Hoshen-Kopelman algorithm. Later, the developed systems were tried to be verified with analyzes such as NIST statistical randomness tests, autocorrelation test, scale-index test. The random bit sequences obtained here are used for encryption processes with the help of existing encryption algorithms and the encryption algorithm that we advanced. The reliability of the encryption systems was investigated by applying different cryptanalysis methods such as histogram analysis, key sensitivity and key space analysis, NPCR and UACI test, entropy test, calculation of correlation coefficients.

In this study, a new symmetric block cipher algorithm is designed with different true and pseudo random number generator designs that meet the security requirements. It has been shown that the random number generators we have developed can be used in cryptological applications by obtaining successful results from statistical tests. On the other hand, the developed symmetric block cipher algorithm, has been shown in our study to work faster than standard algorithms in encryption and decryption processes, as well as succeeding in security tests.

Keywords: Random number generator, Cryptology, Encryption, NIST statistical tests

ŞEKİLLER LİSTESİ

Sayfa

TABLOLAR LİSTESİ

Sayfa

SİMGELER

$A(d)$	: Otokorelasyon değeri
$D(x)$	: Komşu piksellerin korelasyon katsayısı
f	: Keccak algoritma fonksiyonu
H_m	: Bilgi entropisi değeri
i_{scale}	: Scale-index test değeri
PK_n	: Açık anahtar seçimi
S_{obs}	: Bit dizisindeki oran
s_i	: Seçilen bit indisi
T	: Dizideki bitlerin birbirine oranı
$\left\|X5\right\|$	: Bit dizileri için mutlak otokorelasyon testi
$x(m)$	: Elektrik alan sinyali
χ_{obs}	: Bit dizindeki eşit değerli bloklar
V/m	: Metre başına Volt
y_t	: Bit dizisi
$\wedge_{\max}$	: En büyük Lyapunov üsteli
$\Delta(i, j)$	: Piksellerdeki bit değişim formülü

KISALTMALAR

3DES	: Triple Data Encryption Standard (Üçlü Veri Şifreleme Standardı)
AES	: Advanced Encryption Standard (Gelişmiş Şifreleme Standardı)
CMOS	: Complementary Metal-Oxide Semiconductor (Bütünleyici Metal Oksit Yarı İletken)
DES	: Data Encryption Standard (Veri Şifreleme Standardı)
ECG	: Electrocardiogram (Elektrokardiyografi)
EEG	: Electroencephalogram (Elektroensefalografi)
FIPS	: Federal Information Processing Standards (Federal Bilgi İşlem Standardı)
FPAA	: Field Programmable Analog Array (Alanda Programlanabilir Analog Dizi)
FPGA	: Field Programmable Gate Array Alanda Programlanabilir Kapı Dizisi
GRSÜ	: Gerçek Rastgele Sayı Üreteci
HK	: Hoshen-Kopelman Algoritması
HRSÜ	: Hibrit Rastgele Sayı Üreteci
IEEE	: Institute of Electrical and Electronics Engineers (Uluslararası Elektrik Elektronik Mühendisleri Topluluğu)
IP	: İnternet Protokolü
IPI	: Inter-Pulse-Intervals (Darbe Aralığı)
kHz	: Kilo Hertz
LCG	: Linear Congruential Generator (Doğrusal Eşlenik Üreteci)
LFSR	: Linear Feedback Shift Register (Doğrusal Geri Beslemeli Kaydırmalı Kaydedici)
MSE	: Mean Square Error (Ortalama Kare Hatası)
NIST	: National Institute of Standards and Technology (Ulusal Standartlar ve Teknoloji Enstitüsü)
NPCR	: Number of Pixel Change Rate (Piksel Değişim Oranı Sayısı)
MHz	: Mega Hertz
PLL	: Phase Locked Loop (Faz Kilitlemeli Çevrim)
PSNR	: Peak Signal to Noise Ratio (Tepe Sinyal Gürültü Oranı)
RC4	: Rivest Şifreleyicisi 4
RGB	: Red Green Blue (Kırmızı Yeşil Mavi)
RSA	: Rivest-Shamir-Adleman
RSÜ	: Rastgele Sayı Üreteci
SHA	: Secure Hash Algorithm (Güvenli Özet Algoritması)
SRSÜ	: Sözde Rastgele Sayı Üreteci
UACI	: Unified Avarage Changing Intensity (Birleşik Ortalama Değişken Yoğunluğu)
VHDL	: Very High Speed Integrated Circuit Hardware Description Language (Yüksek Hızlı Tümleşik Devreler İçin Donanım Tanımlama Dili)
YSA	: Yapay Sinir Ağları

1. GİRİŞ

Veri güvenliğinin öneminin gün geçtikçe artmasıyla geliştirilen yeni şifreleme yöntemleri de karmaşık hale gelmektedir. Mantıksal ve işlemsel operatöarler, algoritmayı karmaşıklaştırarak veri güvenliğini arttırsa da bazen dezavantaj oluşturabilir. Genel olarak, verilerin gizli kalması, bütünlük içermesi ve doğrulanabilmesi gibi faktörler bilgi güvenliğinin vazgeçilmezleridir. Sistemlerin karmaşık hale getirilmesi ile başkaları tarafından ele geçirilmemesi için uğraşılırken bazen de iletilmek istenilen yere mesajın bozuk ya da eksik şekilde iletilmesi söz konusu olabilir. Bu yüzden şifreleme algoritmaları çok doğru seçilmelidir. Bu alanda yapılan araştırmalar genel olarak, bu odak noktası çevresinde şekillenmiştir. Örneğin günümüzde yaygın olarak kullanılan standart şifreleme algoritması olan AES (Advanced Encryption Standard, Gelişmiş Şifreleme Standardı) şifreleme standardını belirlemek için yapılan yarışmada, AES şifreleme sistemine göre çok daha karmaşık ve verilerin ele geçirilmesini zorlaştıracak algoritmalar da önerilmiştir. Ancak veri doğrulaması yapabilmesi ve şifreleme-şifre çözme işlemlerindeki çabukluğu sebebiyle AES en çok tercih edilen standart olmuştur.

Kriptoloji alanında en önemli parametre, şifreleme anahtarıdır. Bu anahtarın üretilmesi, dağıtılması ve gizlenmesi en büyük problemlerin başında gelir. Kaotik sistemlerin içerdiği karmaşıklık parametreleri, başlangıç şartlarına olan hassas duyarlılığı bu sistemleri anahtar üretmek için kullanılabilir hale getirmiştir. Şifreleme işlemlerinde kullanılan Rastgele Sayı Üreteçlerinin (RSÜ) ürettiği sayıların rastgeleliği, kriptolojik çalışmaların güvenliğini yakından ilgilendirmektedir. Son yıllarda popüler bir araştırma konusu olan kaotik devreler ve RSÜ tasarımları, literatürde yaygın olarak bulunmaktadır. Elde edilen rastgele sayı dizileri, NIST-800-22 ya da FIPS-140-1 gibi uluslararası geçerliliği olan bir dizi testlerden geçtikten sonra şifreleme uygulamalarında kullanıma uygun olup olmadıkları belirlenmektedir [1].

Şifreleme işlemleri yapılırken önemli noktalardan birisi de yazılımın öz kaynaklar ile gerçekleştirilmesidir. Hassas bir işlem olan şifreleme işlemleri sırasında fark edilmesi çok zor olan güvenlik açıkları olabilir. Çeşitli sebeplerden dolayı meydana gelen bu güvenlik açıkları bilgi güvenliğini tehlikeye sokmaktadır. Tez çalışmasında bu yönüyle güvenilir bir şifreleme sistemi oluşturulması için gerekli altyapıyı oluşturmak amaçlanmıştır. Böylece, yerli bir şifreleme sistemi geliştirilmesine yeni bir katkı sunulup bu konudaki farkındalık arttırılacaktır. Sunulacak olan yenilikçi çözümler, kritik uygulamalardaki ihtiyaçları karşılayabilecektir. Önerilen şifreleme sisteminin güvenliği kadar etkinliği de son derece önemlidir. İyi bir şifreleme algoritması, kriptanaliz uygulamalarına karşı gerekli direnci göstermesinin yanı sıra hızlı işlem de yapabilmelidir. Sadece güvenliği yüksek olan şifreleme algoritmasının çok yavaş işlemesi, algoritmanın kullanım alanı bulmasını oldukça zorlaştıracaktır.

Kaotik sistemler temelde ayrık zamanlı ve sürekli zamanlı yapılar olarak ikiye ayrılmaktadır. Her iki sistemin de şifreleme uygulamalarında kullanılabildiği çeşitli araştırmalarla gösterilmiştir. Ancak doğrudan kaotik bir sistem ile elde edilen verilerin, şifreleme de kullanılması her zaman doğru olmayacaktır. Şifrelenmiş bir verinin, çeşitli saldırılara karşı dayanıklı olduğunun kanıtlanması gerekmektedir. Veri güvenliğini gerçek anlamda sağlamak için çeşitli araştırmalar yapmak gerekir. Güçlü olmayan bir şifreleme algoritmasının sisteme işlem yükü getirmekten başka bir katkısı olmayacaktır. Literatürde kabul edilen histogram analizi, diferansiyel saldırı, kaba kuvvet saldırısı ve korelasyon gibi analizlerin yapılması sistemin güvenilirliğini ispatlayacaktır. Bunun yanında sistemin hızı, maliyeti ve bellek kullanımı gibi parametreler göz önüne alınarak optimal bir çözüm sunmak için sistem tasarımı yapılmalıdır [2,3].

1.1. Literatür Özeti

Son yıllarda kaotik ve hiper kaotik sistemler üzerine yapılan araştırmalar geniş uygulama alanları bulunan çalışmaların literatüre sunulmasını sağlamıştır [4–6]. Mevcut sistemlerin özelliklerini geliştiren ya da tamamen yeni bir sistem olarak literatüre sunulan çalışmalar bulunmaktadır [7].

Önceki yıllarda, kaotik modülasyon, kaotik maskeleme ve kaotik anahtarlama gibi kriptolojik işlemler analog olarak yapılsa da artık kaotik tabanlı kriptolojik işlemler genelde sayısal olarak yapılmaktadır. Sayısal olarak yapılan işlemlerin en büyük avantajı şifreleme için kullanılan algoritmanın kolay bir şekilde şifre çözme algoritmasına çevrilebilmesidir. Analog devrelerde şifreleme için kullanılan devreyi tasarlamak kadar şifre çözme sistemini de kurmak zordur. Ayrıca algoritmayı değiştirmek için sayısal sistemlerde küçük bir kod değişikliği yeterli olurken analog sistemlerde çoğu zaman devrenin kısmen veya tamamen yeniden tasarlanması gerekecektir. Ayrıca farklı ortam şartlarından etkilenmesi sebebiyle analog sistemlerin kullanımları çok sınırlıdır [8].

Şifreleme işlemleri için geliştirilen kaotik sistemler, FPGA (Alanda Programlanabilir Kapı Dizisi) sistemi üzerinde uygulanabilmektedir [9–13]. Merah vd. yaptıkları çalışmada FPGA yapısı üzerinde VHDL (Yüksek Hızlı Tümleşik Devreler için Donanım Tanımlama Dili) dilini kullanarak bir bilgi güvenliği sistemi tasarlamıştır [14]. Bao vd. yaptıkları çalışmada, dört temel devre elemanı kullanarak kaotik bir devre tasarlamıştır. Matlab üzerinde simülasyonları gerçekleştirilen devrenin board üzerinde gerçekleştirilmesi de sağlanmıştır. Devrenin kaotik davranış sergilediği grafikler yardımıyla desteklenmiştir [15]. Güler vd. yaptıkları çalışmada, kaotik Chen sistemi senkronizasyonuyla güvenli bir haberleşme sistemi önermiştir. Labview üzerinde görsel arayüzü oluşturulan sistemin gerçek zamanlı olarak kullanımına değinilmiştir [16]. Avaroğlu 2020 yılında yaptığı çalışmada, fiziksel olarak klonlanamayan bir fonksiyonu FPGA üzerinde farklı bir yöntemle gerçekleştirmiştir. Gerçek zamanlı bitler üretilen sistem, NIST testlerinden başarı elde etmiştir [17]. Yakut vd. yaptıkları çalışmada, Gerçek Rastgele Sayı Üreteci (GRSÜ)'ler için yeni güvenli bir

yaklaşım geliştirmiştir. Keccak son-işlem algoritması kullanılan çalışma, FPGA üzerinde gerçekleştirilmiştir. Ham veriler NIST testlerinden başarı elde edemezken, işlenmiş veriler tüm testleri başarıyla geçmiştir [18]. Kim vd. yaptıkları çalışmada, rüzgâr enerjisiyle elektrik üretimi gerçekleştiren sistemin sergilediği kaotik davranıştan faydalanmıştır. 30 psi rüzgâr basıncı altında 4.91 mW/m^2 elektrik enerjisi üreten sistem tüm NIST testlerinden başarı sağlamıştır. Analog-Dijital dönüştürücüdeki örnekleme frekansı değiştirilerek sistemin daha başarılı sonuçlar alabileceği belirtilmiştir [19]. Saravanan ve Sivabalakrishnan yaptıkları çalışmada, hibrit kaotik bir haritadan faydalanarak balina optimizasyon algoritması desteğiyle parametre değişikliğine dayanan bir görüntü şifreleme sistemi geliştirmiştir. Gri tonlamalı resimler üzerinden başarılı sonuçlar veren sistem kriptolojik analizlerde de başarı sağlamıştır [20]. Gong vd. 2021 yılında yaptıkları çalışmada, kesir dereceli Chua sistemlerinden faydalanarak bir kaotik seri üretmiştir. Kriptolojik uygulamalarda başarılı bir şekilde kullanılabilen sistem iyi bir entropi kaynağı sunmaktadır [21].

Kaotik sistemler özellikle GRSÜ tasarımları için fazlaca tercih edilmektedir. Sistem tasarımları kaos yardımıyla oldukça kolaylaşmaktadır [22–31]. Arslan Tuncer yaptığı çalışmada, gerçek zamanlı bir Rastgele Sayı Üreteci tasarlamıştır. Halka osilatör tabanlı olan tasarım, çift fiziksel olarak kopyalanmaz fonksiyon algoritmasıyla desteklenmiştir [32]. Babaei ve Farhadi yaptıkları çalışmada, Sözde Rastgele Sayı Üreteçlerini çeşitli testlere tabi tutarak güvenilir olup olmadıklarını incelemişlerdir. Literatürde bulunan 10 farklı Sözde Rastgele Sayı Üreteci üzerinde yaptıkları çalışmada istatistiksel test olarak NIST ve TestU01'den faydalanmışlardır. Test sonuçlarına bakılarak rastgele sayı üreteçlerinin güvenilirlikleri tartışılmıştır [33]. Guler vd. 2017 yılında yaptıkları çalışmada, CMOS (Bütünleyici Metal Oksit Yarı İletken) tabanlı halka osilatör yardımıyla rastgele sayı üreteci tasarlamışlardır. Entropi analizi yapılan sistem için farklı halka sayıları değerlendirmeye alınmıştır. 3.02 Mbps maksimum bit üretim hızı elde edilmiştir [34]. Buchovecka vd. halka osilatörleri yardımıyla kurulmuş, fiziksel olarak klonlanamaz devre parçası yardımıyla gerçek rastgele sayı üreteci gerçekleştirmişlerdir. NIST testi, uygulanan rastgele sayı üreteci çıktılarına Von-Neumann ve XOR son-işlem algoritmaları da ilave edilmiştir. Son-işlem uygulamalarının gerekliliği savunulan çalışmada bu uygulamaların çeşitli oranlarda rastgele sayı üreteci çıktılarını azalttığı gösterilmiştir [35]. Bhattacharjee vd. geliştirdikleri hücresel otomat sistemiyle, sözde rastgele sayı üreteci tasarımı yapmıştır. Lineerlik açısından incelenen sistemin başarımı değerlendirilerek C dilindeki kodları sunulmuştur [36].

GRSÜ tasarımları yapılırken EEG (Elektroensefalografi) ve ECG (Elektrokardiyografi) gibi elektriksel sinyallerin yanı sıra farklı gerçek sinyal kaynaklarından da faydalanılmaktadır [37–41]. Abutaleb yaptığı çalışmada, quatum hücresel otomatlardan faydalanarak gerçek rastgele sayı üreteci tasarımı gerçekleştirmiştir. NIST testlerinden başarılı olan nano teknoloji sisteminin performansı başarılı bulunmuştur [42]. Ramakrishnan yaptığı çalışmada, kriptolojik teknikleri detaylı bir şekilde değerlendirmiştir. Biyometrik verilerle şifreleme tekniklerine de değinilen

çalışmada farklı tekniklere değinilmiştir [43]. Rai vd. tasarladıkları rastgele sayıcı üretecini NIST testlerinden başarılı sonuçlar elde ettiğini yaptıkları çalışmayla göstermiştir. Memristör kullandıkları çalışmada CMOS cihazlara göre çok daha az enerji tüketimi meydana geldiği belirlenmiştir. Önerilen sistemin literatüre göre rastgelelik konusunda daha iyi çıktılar elde ettiği gösterilmiştir [44]. Alhadawi vd. geliştirdikleri sözde rastgele sayı üreteci tasarımında, LFSR (Doğrusal Geri Beslemeli Kaydırmalı Kaydedici) ve ayrık kaotik haritalar kullanmıştır. Performans analizi yapılan gerçek rastgele sayı üretecinin NIST, TestU01 ve DIEHARD istatistiksel rastgelelik testi sonuçlarına yer verilmiştir. Sistemin bazı siber saldırı türlerine karşı dirençli olduğu belirtilmiştir [45]. Coşkun vd. geliştirdikleri rastgele sayı üreteci tasarımında, bilgisayar kontrollü bir analog-dijital dönüştürücü platform kullanmıştır. PIC mikro kontrolörü üzerinde gerçekleştirilen çalışma Zhongtang kaotik sistemini kullanmaktadır. 6 farklı entropi kaynağı kullanılarak tekrarlanan sistemde geliştirilen rastgele sayı üreteçlerinin tamamı NIST testlerinde başarılı olmuştur [46].

Guo ve Zhou yaptıkları çalışmada, çok boyutlu bir kaotik sistem yardımıyla gerçek rastgele sayı üreteci tasarımı yapmıştır. Matlab üzerinde simülasyonu yapılan sistemin LFSR kullanılarak ve kullanılmadan FIPS-140 testlerindeki başarımı değerlendirilmiştir. Örnekleme frekansı olarak 1 MHz seçilmiştir [47]. Elmanfaloty ve Abou-Bakr yaptıkları çalışmada, tek boyutlu kaotik sistem yardımıyla sözde rastgele sayı üreteci tasarlamıştır. Farklı kontrol parametreleri için uygulanan istatistiksel rastgelelik testlerine göre en uygun sistem tasarımı gerçekleştirilmiştir. FPGA gerçeklemesi de yapılan sistemin başarılı sonuçlar elde ettiği gösterilmiştir [48]. Di Patrizio Stanchieri vd. yaptıkları çalışmada, gerçek rastgele sayı üreteci tasarlamıştır. PLL (Faz Kilitlemeli Çevrim) osilatörü ve Flip-Flop'lar yardımıyla kurulan devre FPGA üzerinde gerçeklenmiştir. Sistem NIST testlerinden başarıyla geçerek 100 Mbps bit üretim hızı elde etmiştir [49].

Sözde Rastgele Sayı Üreteci (SRSÜ) tasarımları yapılırken genel manasıyla belirli algoritmalardan faydalanılarak tasarımlar oluşturulmaktadır [50–53] . Chen vd. hiperkaotik bir sistem yardımıyla yeni bir SRSÜ geliştirmiştir. Geliştirilen SRSÜ görüntü şifreleme alanına uygulanmıştır. Kripto-analizi yapılan sistem NIST ve ENT testlerinde başarı sağlamıştır [54]. Li vd. yaptıkları çalışmada, MARC-BB algoritmasını kullanarak SRSÜ tasarımı gerçekleştirmiştir. Saldırılara karşı direnci ve istatistiksel analizi yapılan tasarımın bazı düzenlemeler yapılarak GRSÜ olarak da kullanılabileceği gösterilmiştir [55]. Bhattacharjee vd. 2018 yılında yaptıkları çalışmada, SRSÜ çalışmalarını detaylı bir şekilde incelemeyi amaçlamıştır. 12 farklı alanda sınıflandırılan SRSÜ çalışmalarının performans değerlendirmesi yapılmıştır [56]. Pasqualini ve Parton yaptıkları çalışmada, otomatik bir şekilde farklı SRSÜ'ler oluşturan bir sistem tasarlamıştır. Her seferinden kullanılan algoritmanın farklı olduğu öne sürülen çalışmada derin öğrenmeden faydalanılmıştır [57]. Özkaynak yaptığı çalışmada, yine bir hibrit rastgele sayı üreteci tasarlamıştır. Benzer şekilde

GRSÜ çıktılarını SRSÜ'ye ek girdi olarak kullanan sistem kriptolojik uygulamalarda kullanılabilir olarak değerlendirilmiştir [58].

Tasarlanan rastgele sayı üreteçlerinin şifreleme anahtarı olarak kullanılması üzerine yapılan birçok kriptolojik uygulama bulunmaktadır. Üretilen sayıların rastgeleliği, şifreleme uygulamalarındaki güvenilirliği doğrudan etkilemektedir [59-73]. Hibrit bir yöntemle tasarlanan gerçek zamanlı sistem, yüksek boyutlu verilerin (ses, resim, video) şifrelenmesinde kullanılabilirliği oldukça düşüktür. Yardım ve Afacan yaptıkları çalışmada, şifreleme uygulamalarında kullandıkları kaotik sistem üzerinde gecikme ve anahtarlama işlemleri uygulamıştır [74]. Bu yöntemde en önemli nokta, verilerin hangi sırayla şifrelendiğini bilmek gerekliliğidir. Sohby ve Shehata ise; Lorenz kaotik sistemiyle veriyi şifreleyerek bir uygulama geliştirmiştir [75]. Akgul vd. yaptıkları çalışmada, üç boyutlu ve denge noktası bulunmayan bir kaotik sistemi şifreleme uygulamaları için kullanılabilecek bir tasarıma dönüştürmüştür [76]. Prabu vd. ayrık zamanlı kaotik bir sistem olan lojistik haritalardan faydalanarak bir şifreleme uygulamasını gerçek zamanlı olarak tasarlamıştır [77]. Sahin vd. yaptıkları çalışmada, memristif bir kaotik sistemin hem simülasyon hem de gerçek uygulamasını gerçekleştirerek haberleşme sistemlerinde kullanımını incelemiştir. Farklı multimedya verileri üzerinde geliştirdikleri sistemi denemiştir [78].

Khalil yaptığı çalışmada, ses sinyalleri için gerçek zaman bir şifreleme ve şifre çözme sistemi tasarlamıştır. RSA şifreleme algoritmasını kullanan sistem şifrelenen veriyi bir kanal üzerinden alıcıya aktarılan şifreyi çözerek ses iletimini sağlamaktadır. Matlab üzerinde modellenen sistem başarılı bir şekilde veri akışı sağlamaktadır. Önerilen sistemin ses verisi dışında başka veri tiplerine de uygulanabileceği belirtilmiştir [79]. Shu vd. su altı haberleşme sistemleri için güvenli bir haberleşme platformu tasarlamışlardır. Kaos tabanlı olan sistemin performans analizi yapılarak güvenilirliği gösterilmiştir. Önerilen sisteminin kolay ve pratik bir haberleşme sistemi olduğu belirtilmiştir [80]. Moosavi vd. 2017 yılında yaptıkları çalışmada, kriptolojik çalışmalarda kullanılmak üzere anahtarlar üreten bir sistem tasarlamıştır. Elektro-kardiyografi sinyallerinden faydalanılarak gerçekleştirilen sistem IPI (Darbe Aralığı) tabanlı anahtar üretme sistemini kullanmaktadır. Çalışmada AES şifreleme algoritmasından faydalanılmıştır. Rastgeleliğin tespiti için NIST testlerinden faydalanılmıştır [38].

Khan vd. geliştirdikleri görüntü şifreleme sisteminde, Fibonacci serisi ve ayrık dinamik sistemlerden faydalanmıştır. RGB formatındaki resimleri Kırmızı, Mavi ve Yeşil bileşenler için ayrı ayrı şifreleyerek temelde üç farklı şifrelenmiş görüntü oluşturmuşlardır [81]. Sun vd. 2020 yılında yaptıkları çalışmada, memristif kaotik bir sistem yardımıyla görüntü şifreleme uygulaması yapmıştır. Farklı bir algoritma yardımıyla kaotik sistemin kriptolojik uygulamada kullanıldığı çalışma PSpice üzerinde modellenmiştir [82]. Fang vd. yaptıkları çalışmada, güvenli bir kaotik blok

şifreleme sistemi geliştirmiştir. Sistem görüntü şifreleme üzerine test edilirken gen kodlama bölgesi algoritmalarından faydalanılmıştır. Yapılan çalışma NIST testlerinden başarı sağlamıştır [83].

Alanda Programlanabilir Kapı Dizileri (FPGA) rastgele sayı üreteci uygulamaları için tercih edilen bir çalışma konusu olmuştur [84–93]. Farklı olarak Çiçek ve arkadaşları, FPAA (Alanda Programlanabilen Analog Dizi) donanımı yarımıyla 1.5 Mbps üretim hızında RSÜ tasarlayarak tüm NIST testlerinden başarılı sonuçlar elde etmiştir [94]. Diğer bir çalışmada, Kohlbrenner vd. yaptıkları gerçek rastgele sayı üreteci tasarımında, klasik bir halka osilatör tasarımı kullanmıştır. Bu devrede oluşan seğirmeden faydalanan araştırmacılar sinyalleri örnekleyerek tasarımı oluşturmuştur. XOR son-işlem uygulamasına yer verilen çalışmada 1 Mbps'den daha az bir sayı üretim hızı tespit edilmiştir. Sistem tüm NIST testlerinden başarı elde etmiştir [95]. Bu sistemin etkinliğini göstermek için Sunar tarafından önerilen halka osilatörlü gerçek rastgele sayı üreteci kullanılmıştır. Burada uygulanan son-işlem algoritması normalde uygulanan Resilient algoritmasına göre performans artışı sağlayarak 2.5 Mbps olan bit üretim hızını 20 Mbps seviyesine çıkarmıştır [96]. Bu ve bunun gibi birçok sistemde kullanılan halka osilatörler uygulama kolaylığı, tüm tasarımın FPGA gibi çipler içerisinde yer alıyor olması ve sabit çıkış hızına sahip olmaları gibi avantajlarıyla literatürde sıklıkla tercih edilen sistemlerin başında gelmiştir. Ancak halka osilatörlerin fazlaca kullanılması yüksek güç tüketimi ve düşük bit üretim hızı gibi dezavantajlar ortaya çıkarmaktadır. Ayrıca halka osilatör yöntemiyle üretilen rastgele sayıların genellikle düşük korelasyona sahip olması sebebiyle son-işlem algoritmalarına ihtiyaç duymaları önemli bir parametredir.

Son yıllarda gelişmeye başlayan FPGA tabanlı GRSÜ sistemleri temel alarak SRSÜ'lerde meydana gelebilecek güvenlik açıklarını gidermeyi amaçlayan sistemler yaygınlaşmıştır. FPGA tabanlı sistemler hem donanım tabanlı oldukları için hem de yüksek çalışma hızları sundukları için tercih edilebilir olmuşlardır. SRSÜ'lerde bulunan başlangıç değerlerini ya da bazı parametreleri GRSÜ'lerden alarak tasarlanan sistemler hem yüksek entropi hem de uygulama kolaylığı ve yüksek bit üretim hızı sunabilmektedir [97–102].

FPGA tabanlı bazı çalışma örneklerinde YSA (Yapay Sinir Ağları) ya da GA (Genetik Algoritma) gibi yöntemlerden faydalanılarak optimizasyon işlemi uygulanan RSÜ tasarımları da ortaya çıkmıştır [103–108]. Bu uygulamalar yardımıyla optimizasyon yöntemlerinin RSÜ'ler için önemli bir gelişim alanı olduğu görülmüştür.

1.2. Problemin Tanımlanması ve Tezin Amacı

Rastgele Sayı Üreteçlerinin istatistiksel örnekleme çalışmaları, şans oyunları, bilgisayar simülasyonları, kriptografi, rastgele sistem tasarımları gibi tahmin edilebilirlik faktörünün oldukça zayıf olması istenen birçok kullanım alanı vardır. Her ne kadar literatürde çok miktarda rastgele sayı üreteci mevcut olsa da geniş kullanım alanı içerisinde sürekli yeni sistemlere ihtiyaç duyulması

sebebiyle günümüzde hala RSÜ tasarımları popülerliğini korumaktadır. Ayrıca gelişen teknoloji sayesinde RSÜ sistemlerinin gün geçtikçe daha az sistem gereksinimiyle daha hızlı sayı üretimini gerçekleştirmesi mümkündür.

Şifreleme sistemleri içerisinde RSÜ sistemlerinin kullanıldığı bilinmektedir. Ancak son yıllarda literatüre giren özellikle gelişmiş görüntü ve ses şifreleme çalışmaları incelendiğinde RSÜ sistemleriyle bağlantıları açık şekilde belirtilmemektedir. Yine gerçekleştirilen yeni RSÜ tasarımlarının birçoğunun çeşitli analizler yardımıyla şifreleme sistemleri için uygun olduğu gösterilse de tam olarak ne şekilde şifreleme sistemleri içerisinde kullanılacakları gösterilmemiştir [22,109,110].

Bu tez çalışmasında, yeni rastgele sayı üreteci tasarımları ve şifreleme uygulamaları içerisindeki kullanımları incelenmiştir. GRSÜ ve SRSÜ niteliği taşıyan farklı rastgele sayı üreteçlerinin avantaj ve dezavantajları gösterilmeye çalışılmıştır. Geliştirilen rastgele sayı üreteçlerinin çeşitli analizler yardımıyla kriptografik uygulamalar için uygunluğu kanıtlanmaya çalışılmıştır. Daha sonra literatürde mevcut olan algoritmalarla uygulamalar yapılarak, rastgele sayı üreteçleri bu alanlarda kullanılmaya çalışılmıştır. Tez çalışmaları kapsamında geliştirilen simetrik blok şifreleme algoritmasıyla, şifreleme uygulamaları yapılarak sistemin güvenlik analizleri gerçekleştirilmiştir. Geliştirmiş olduğumuz şifreleme algoritması tüm veri tipleriyle uyumlu olarak çalışabilse de görüntü şifreleme alanının bazı avantajları sebebiyle bu alana yoğunlaşılmıştır. Bu avantajlar şu şekilde sıralanabilir:

1) Görüntü iletiminin yaygınlığı,
2) Görüntü şifreleme sistemleri için detaylı güvenlik analizlerinin bulunması,
3) Görsel olarak şifreleme sisteminin sunumunun daha kolay olması.

Tüm bu nedenlerle literatürdeki şifreleme uygulamaları genellikle görüntü üzerine odaklanmıştır [111–117]. Geliştirilen şifreleme algoritması bu sebeplerle görüntü şifreleme işlemlerinde kullanılarak analizleri yapılacaktır. Şifreleme sistemi içerisinde yeni geliştirilen özgün RSÜ tasarımlarının görüntü karıştırma, şifreleme anahtarı üretme ve şifreleme anahtarlarını seçme işlemlerinde ne şekilde kullanılabileceği açıklanacaktır. Bu şekilde literatürde bulunan bir eksikliğin hem işlem hızı hem de uygulanabilirlik açısından avantajlı bir sistem yardımıyla giderilmesi amaçlanmaktadır.

1.3. Tezin Yapısı

Bu tez çalışmasının genel yapısının toplamda altı bölümden oluşması planlanmaktadır. İlk bölüm olan giriş bölümünde tezle ilgili genel bilgiler verilmiştir. Bu bölümde bilimsel problemin ortaya konulması, problemin önemi ve tez çalışmasının amacı gibi temel konulara değinilmiştir. Problemle ilgili hipotezler yine bu bölümde açıklanmıştır. Literatürde bulunan diğer çalışmalar ile tezde yapılan çalışmanın temel detayları da paylaşılmıştır.

İkinci bölümde, tez çalışmasının temelini oluşturan rastgele sayı üreteçlerinden bahsedilmiştir. RSÜ çeşitleri ve özellikleri bu bölümde açıklanmıştır.

Üçüncü bölümde, şifreleme sistemlerinin genel yapısı açıklanmıştır. Temel anlamda hangi şifreleme çeşitlerinin bilgi güvenliği amacıyla kullanıldığı ve özellikleri belirtilmiştir.

Dördüncü bölümde tez çalışmasında kullanılan materyal ve metottan bahsedilmiştir. Hangi yöntemlerle rastgele sayı üretecinin tasarlandığıyla başlayan bölüm RSÜ analiz ve değerlendirme yöntemleriyle devam etmiştir. Daha sonra önerilen şifreleme sistemlerinden bahsedilmiştir. Bu şifreleme sistemlerinin güvenlik analizlerinin ne şekilde yapıldığı açıklanmıştır.

Beşinci bölümde tez çalışmasında elde edilen sonuçlar paylaşılmıştır. Rastgele sayı üreteçlerinin istatistiksel testlerinin sonuçları ve şifreleme uygulamaları da bu bölümde gösterilmiştir. Şifreleme uygulamaları sonucunda elde edilen çıktılar, kriptanaliz yöntemleriyle değerlendirilerek sistemin başarısı belirtilmiştir.

Altıncı bölümde tez çalışmasının genel değerlendirilmesi yapılmıştır. Tez çalışması çıktısının ne anlam ifade ettiği ve bu çalışmaların hangi alanlarda kullanılabileceği belirtilmiştir. İleride yapılabilecek çalışmalar hakkında, bu bölümde bilgiler verilmiştir.

2. RASTGELE SAYI ÜRETEÇLERİ

Rastgele sayı üreteçleri genel olarak iki ana başlıkta incelenir. Sözde Rastgele Sayı Üreteçleri yazılımsal olarak rastgele sayı üretimi yaparken gerçek rastgele sayı üreteçleri donanıma ihtiyaç duymaktadır. Gerçek Rastgele Sayı Üreteçleri analog ve sayısal tabanlı olarak ikiye ayrılmaktadır. Sürekli zamanlı sistemler analog rastgele sayı üreteçlerini kapsarken, ayrık zamanlı sistemler sayısal rastgele sayı üreteçlerini kapsamaktadır.

Rastgele sayı üreteçlerinde kullanılan önemli bir devre türü de kaotik tabanlı devrelerdir. Başlangıç şartlarına hassas bağımlı ve karmaşık davranışları nedeniyle rastgele sayı üreteçleri içerisinde kolaylıkla kullanılabilmektedir. Ancak kaotik sistem çıkışı olarak alınan rastgele dizilerin bit yapısında bulunmaması bir ön işlem gereksiniminin olduğunu göstermektedir. Öncelikle istenilen RSÜ yapısına göre bit düzeyinde işlemler tercih ediliyorsa, elde edilen sayıların ikili sayı formatına dönüştürülmesi gerekmektedir. İkili sayı formatına dönüştürülen rastgele diziler analiz edilirken daha hassas sonuçlar elde edilmesi muhtemeldir. Burada elde edilen rastgele sayı dizilerinin başarımlarını test edebilmek için NIST-800-22, DIEHARD ya da TestU01 gibi istatistiksel rastgelelik testleri uygulanmalıdır. Elde edilen test çıktılarına göre gerçekleştirilen tasarımlar tekrar gözden geçirilebilir. Test takımları içerisinde genellikle birçok farklı test bulunmaktadır. Bu testlerin her biri rastgele sayı dizisinin farklı özelliklerini inceler ve bütünlük arz ederler. Yani test uygulanan rastgele bit dizisi eğer bir testten bile başarısız oluyorsa bu bütün güvenilirlik kriterlerini etkileyecektir. Sadece belli bir kısım testlerden başarı elde edilmesi, rastgele sayı dizisinin başarılı olduğunu göstermez. Rastgele sayı dizisinin bazı istatistiksel özellikleri arttırılmak isteniyorsa test sonuçlarına bakılmaksızın son işlem algoritmaları uygulanabilir [22]. Ayrıca uygulanan bu algoritmalar istatistiksel testlerde başarı sağlayamayan RSÜ sistemleri için de kullanılabilir.

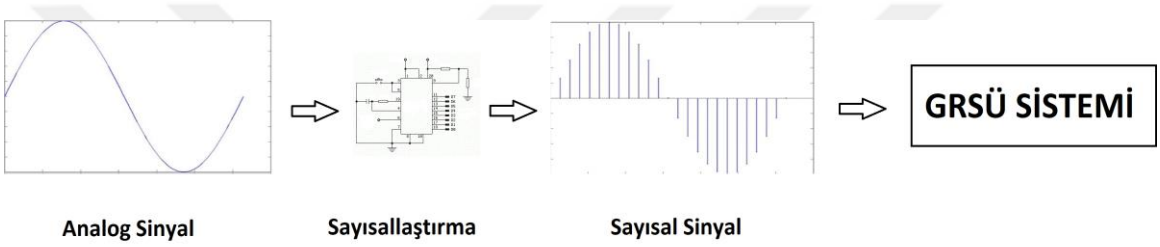
Kaotik bir sistemden ya da ölçüm verilerinden elde edilen sayılar, farklı yapılarda bulunabilirler. Rastgele sayı üreteçleri öncelikle bu sayılara normalizasyon işlemi uygulayarak ikili sayıya dönüştürmektedir. Daha sonra son işlem algoritmaları kullanılarak standart bir rastgele sayı üreteci elde etmek mümkündür. İstatistiksel testlerden başarı sağlayan rastgele sayı üreteçleri uygulama alanlarında kullanım için kabul görmektedir. Şifreleme uygulamalarında kullanılan anahtarların rastgele olması en önemli parametrelerden birisi olarak görülmektedir. Bu alanda yapılan yeni rastgele sayı üreteci tasarımları literatür içerisinde yer almıştır. Çok yüksek frekanslarda çalışan ve düşük güç tüketen sistemler genellikle tercih sebebi olmuştur. RSÜ çıktıları farklı farklı sistemlerde şifreleme anahtarı olarak kullanıldığında çok daha güvenilir oldukları belirtilmiştir [28,76].

Rastgele sayı üreteci tasarımlarının güvenilirliğini ispatlamak için ilk şart istatistiksel testlerdir. Bu testlerden başarılı bir şekilde geçemeyen tasarımların kriptolojik uygulamalarda

kullanılması veri güvenliğini tehlikeye atacaktır [19,95,118]. Daha sonra uygulama alanına göre, çeşitli parametrelerle yeniden üretilebilen veya üretilemeyen RSÜ'ler tasarlanabilir. Aynı rastgele sayı dizilerini farklı ortamlarda kullanmak isteyen sistemler yeniden üretilebilen RSÜ'leri tercih etmektedir. Ancak tek seferlik şifreleme sistemlerinin tasarımında tekrar üretilmesi mümkün olmayan rastgele sayı dizileri kullanılır.

2.1. Gerçek Rastgele Sayı Üreteçleri

Gerçek Rastgele Sayı Üreteçleri, isminden de anlaşıldığı gibi gerçek bir sinyal temelinde oluşturulan rastgele sayı üreteçleridir. Şekil 2.1.'da analog ve sayısal sinyallerin GRSÜ sistemi içindeki kullanım şekilleri gösterilmiştir.



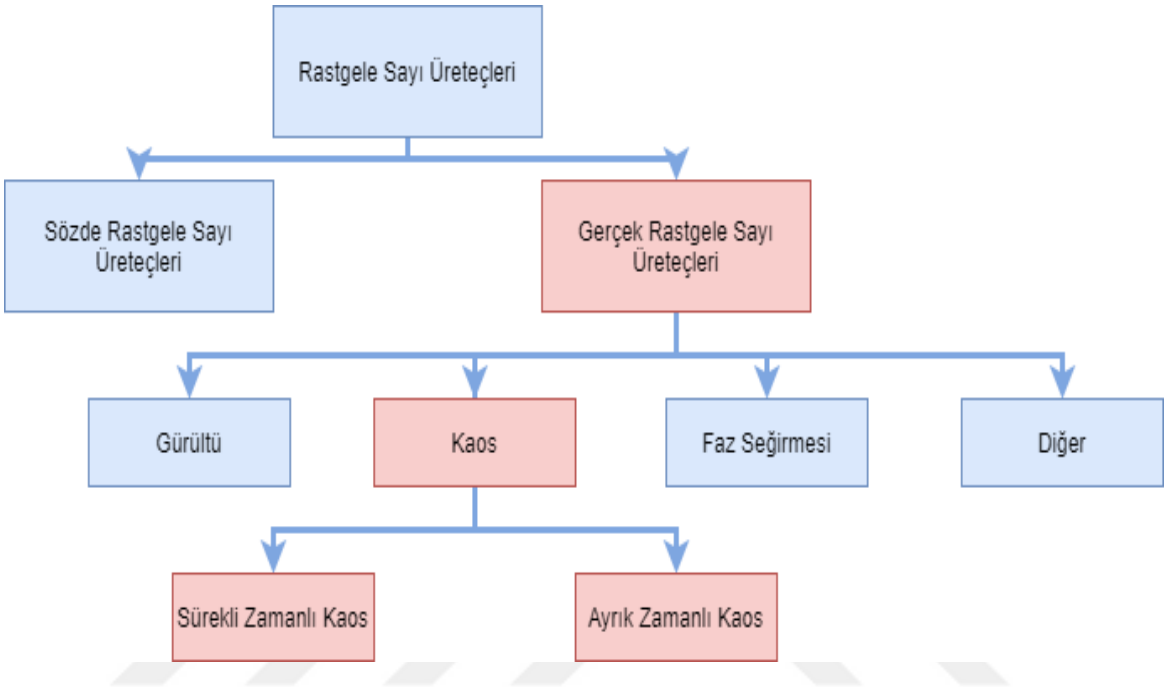
Şekil 2.1. GRSÜ sisteminde analog sinyal ve sayısal sinyal kullanımı

GRSÜ'ler genellikle gerçek dünya sinyallerinden alındığı için bir sayısallaştırma işlemine tabi tutularak rastgele sayı üretiminde kullanılırlar. Ancak doğrudan sayısal sinyallerle de GRSÜ tasarlamak mümkündür. GRSÜ'lerin en büyük özelliği bir entropi kaynağından verileri sağlayarak tekrarsız istatistiksel olarak iyi özelliklere sahip olmasıdır. Örneğin, telegraf gürültü sinyaliyle GRSÜ tasarlamak mümkündür [119]. Bunun gibi farklı rastgelelik kaynakları GRSÜ içerisinde kullanılarak farklı özelliklere sahip rastgele sayı üreteçleri oluşturulabilir [120–126]. Teh vd. yaptıkları çalışmada, ses sinyallerini kullanarak bir GRSÜ tasarlamıştır. Yapılan çalışmada farklı son-işlem algoritmalarının etkileri incelenmiştir [127]. Hu vd. bilgisayar arayüzü olan fare yardımıyla bir GRSÜ tasarımı gerçekleştirmiştir. Ayrıca geliştirilen sistem kaotik bir kriptografi işleminde kullanılmıştır [128]. Burri vd. tekli foton görüntü sensörü tabanlı bir GRSÜ geliştirmiştir. Geliştirilen sistemin literatürdeki çalışmalara göre çok daha az enerji tükettiği belirtilmiştir [129]. Polack yaptığı çalışmada, konser salonlarında oluşan gürültüden rastgele sayı üretilip üretilemeyeceğini araştırmıştır [130].

Yapılan çalışmalardan görüldüğü gibi, entropi kaynağı olabilecek her türlü sinyal ve gürültü yardımıyla GRSÜ tasarlamak mümkündür. Bu alanda yapılan çok miktarda çalışma olmasına rağmen halen farklı rastgelelik kaynakları bularak sayı üreteçleri oluşturulmaktadır. Sinyalin

sayısallaştırılmasından sonra gerekli ise son-işlem algoritmalarının uygulanması kabaca bir GRSÜ tasarımını tanımlamaktadır.

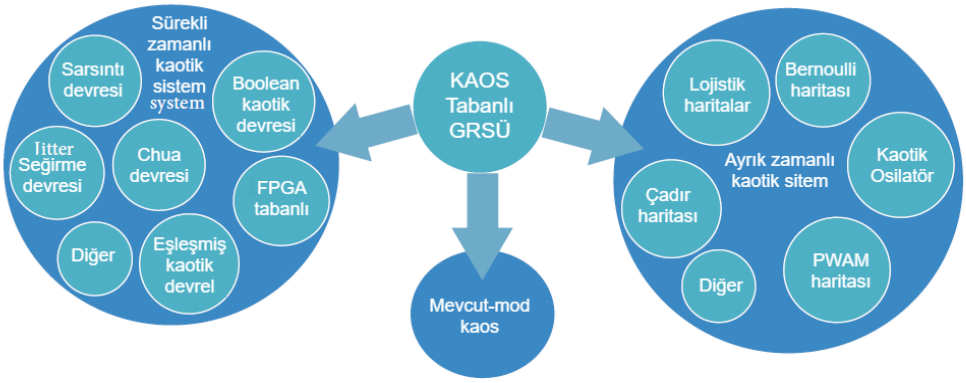
Şekil 2.2'de kaotik sistemlerin rastgele sayı üreteçleri içindeki yeri gösterilmiştir. Kaotik sistemler, GRSÜ çalışmalarındaki devre tasarımları içerisinde geniş bir yer tutmaktadır [84].



Şekil 2.2. Rastgele sayı üreteci çeşitleri ve kaosun yeri

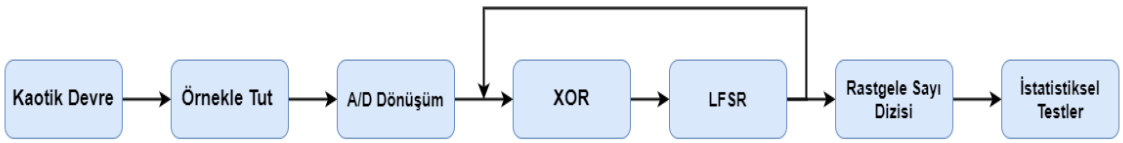
Literatürde kaos temelli rastgele sayı üreteçleri için önerilen çalışmalar incelendiğinde; Chen ve Ueta, geliştirdikleri sistemde Lorenz kaotik sisteminden esinlenerek, Chen kaotik sistemini geliştirmiştir [131]. Lu ve Chen ise, Lorenz ve Chen kaotik sistemini temel alarak tamamen yeni bir kaotik sistem geliştirmiştir [132]. Sprott, detaylı çalışmaları sonucunda 19 farklı sistemi birden tasarlamıştır [133]. Kaotik sistemlerin uygulama alanlarına göre farklı özelliklere sahip olması beklenebilir. Örneğin; kriptolojik uygulamalar için, gizli çekicili kaotik istem adıyla bilinen, denge noktası bulunmayan kaotik çalışmalar mevcuttur [134].

Şekil 2.3'de kaos tabanlı sistemler yardımıyla oluşturulan rastgele sayı üreteci çeşitleri verilmiştir. Farklı ayrık zamanlı ve sürekli zamanlı kaotik sistemler yardımıyla GRSÜ tasarımı yapılabileceği gösterilmiştir.



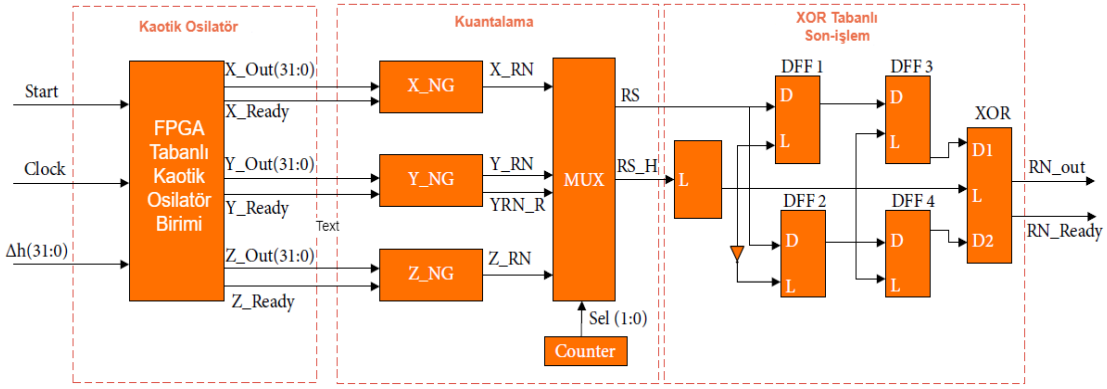
Şekil 2.3. GRSÜ tabanlı sistemlerde kaotik devrelerin kullanımı [135]

Kaya yaptığı çalışmada Chua ve halka osilatörlü fiziksel olarak klonlanamayan fonksiyon yardımıyla bir gerçek rastgele sayı üreteci meydana getirmiştir. Gerçekleştirilmiş Chua devresinden topladığı verileri öncelikle normalize eden sistem daha sonra halka osilatörlü fiziksel olarak klonlanamayan fonksiyonla rastgele sayı üreteci tasarımını tamamlamaktadır. Farklı rastgelelik testleriyle değerlendirilen sistem tüm testlerden başarıyla geçmiştir [110]. Chua devresi tabanlı gerçek rastgele sayı üreteci için blok diyagramı Şekil 2.4'te verilmiştir.



Şekil 2.4. Chua devresi tabanlı gerçek rastgele sayı üreteci blok diyagramı [136]

Avaroğlu yaptığı tez çalışmasında AES standardına göre üretilen sözde rastgele sayı üreteci tasarımına Xilinx FPGA çipi üzerinde tasarlanan Burke-Shaw kaotik sistemini girdi olarak ekleyerek daha yüksek güvenlikli bir sistem gerçekleştirmiştir. Oluşturulan sistem Burke-Shaw kaotik sisteminden çıkan 128 bitlik veriyi AES sisteminden elde edilen 128 bitle XOR işlemine tabi tutarak çalışmaktadır. Bu çıktı yine sisteme geri verilerek iç durum değeri olarak kullanılmış ve farklı bit dizileri elde edilebilmiştir. Rastgele sayı üreteci IEEE 754-1985 standartlarına uygun olan FPGA çipinde kodlanırken RK5-Butcher algoritmasından faydalanmıştır. Sistem çalışma frekansı olarak 373 MHz sunmuştur. Geliştirilen bu hibrit rastgele sayı üreteci tüm NIST testlerinden başarı sağlamıştır [137]. FPGA tabanlı rastgele sayı üreteçlerinin genel şeması Şekil 2.5'te verilmiştir.



Şekil 2.5. FPGA tabanlı rastgele sayı üreteçlerinin genel blok şeması [84]

FPGA tabanlı sistemler öncelikle GRSÜ tasarımlarındaki ortalama bit üretim hızını arttırmayı amaçlamaktadır. Analog elemanlarla gerçekleştirilen devrelerin çalışma frekansları FPGA tabanlı sistemlere göre daha düşük olmaktadır. Ayrıca tümleşik devrelerin sistem parametrelerinin değiştirilmesinin zor olması da FPGA tabanlı sistemlerin önünü açmaktadır [102].

2.2. Sözde Rastgele Sayı Üreteçleri

Sözde Rastgele Sayı Üreteçleri, gerçek bir veri kaynağına dayanmayan genellikle bir algoritma ya da yazılım tarafından üretilen rastgele sayı dizileridir. Bir veri kaynağına ihtiyaç duymaması sayesinden kompakt sistemlerde tercih edilen kullanımı basit rastgele sayı üreteçleridir. En büyük özellikleri oldukça hızlı bit üretimi gerçekleştirebilme imkânına sahip olmalarıdır.

SRSÜ'ler hızlı ve kolay gerçekleştirilebilseler de güvenlik konusundan önemli sorunlar içerebilen uygulamalardır. Belirli bir algoritmaya bağımlı çalıştıklarında yüksek periyotlarla da olsa tekrar etme, tahmin edilebilir olma gibi özellikler içerdiklerinden yüksek güvenlik gerektiren uygulamalarda kullanımları kısıtlıdır.

Literatürde SRSÜ'ler için çok farklı tasarımlar mevcuttur [41,138–140]. Akgül vd. rastgele sayı üreteçleri için bir ara yüz tasarlamışlardır. İstatistiksel testlerde başarılı olan sistem Raspberry Pi üzerinde gerçekleştirilmiştir [141]. Ayubi vd. fraktal tabanlı bir SRSÜ tasarlamıştır. Deterministik kaos oyununa dayalı SRSÜ kriptolojik uygulamalarda başarılı sonuçlar elde etmiştir [142]. Lambic ve Nikolic gerçekleştirdikleri çalışmada kaotik harita tabanlı bir SRSÜ tasarlamıştır. TestU01 ile detaylı bir test sürecine tabi tutulan SRSÜ başarılı sonuçlar elde etmiştir [143]. Gholipour vd. yaptıkları çalışmada, yeni bir SRSÜ tasarlamıştır. Keccak algoritmasını temel alan çalışma hem hızlı hem de istatistiksel olarak başarılı sonuçlar elde etmiştir [144]. Tüm bu araştırmalar incelendiğinde, SRSÜ tasarımları için farklı algoritmalardan faydalanarak çalışmaların yapıldığı görülmektedir. İstatistiksel testlerden başarı elde eden çalışmalar literatürde kendilerine kullanım alanı bulmaktadır.

2.3. Hibrit Rastgele Sayı Üreteçleri

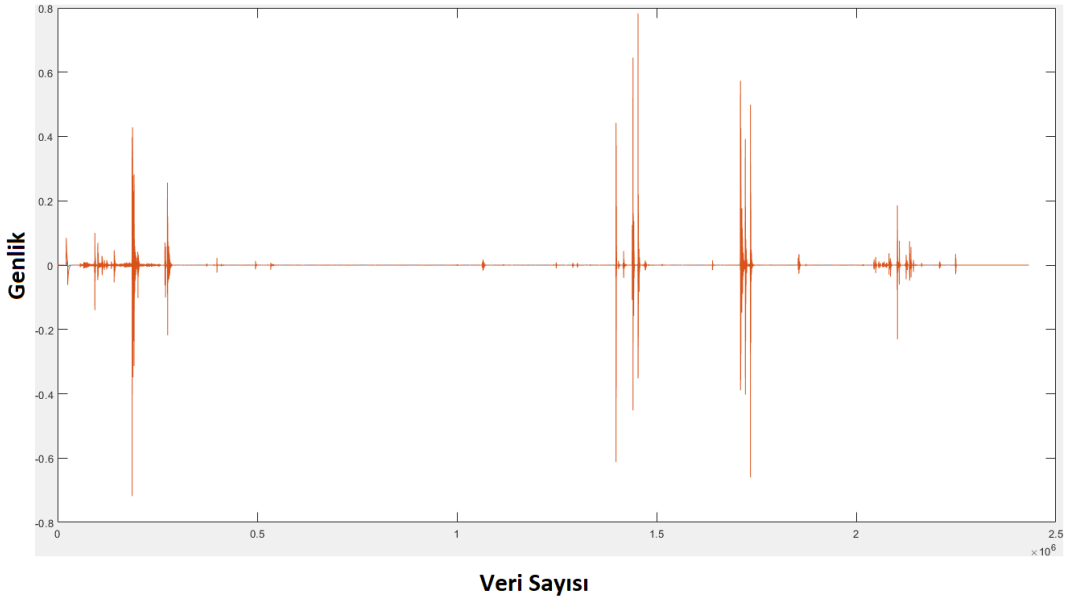
Hibrit Rastgele Sayı Üreteçleri (HRSÜ), hem gerçek hem de sözde rastgele sayı üreteçlerinin taşıdığı özellikleri birleştirerek ortaya çıkan yapılardır. Yani, rastgele sayı üreteci içerisinde hem gerçek bir sinyale dayalı veri, hem de algoritmik işlemler bulunmalıdır. Genellikle GRSÜ'lerin çıktılarına çeşitli son-işlem algoritmalarının uygulanması HRSÜ'lerin oluşumuna sebep olur.

Yakut ve Özer yaptıkları çalışmada, hibrit bir rastgele sayı üreteci tasarlamıştır. Testlerden başarıyla geçen sistem Keccak algoritmasını son-işlem olarak kullanmıştır [109]. Bu örnek çalışmada da göründüğü gibi normalde tasarlanan GRSÜ bir son işlem algoritmasına tabi tutularak HRSÜ'ye çevrilmiştir. GRSÜ'ler genellikle iyi istatistiksel özelliklere sahip olsalar da son-işlem algoritmalarının uygulanması istatistiksel özellikleri önemli ölçüde iyileştirdiği birçok çalışmada kanıtlanmıştır [18,32,96,99]. Son-işlem algoritmalarının çalışma hızı genellikle GRSÜ sistemlerin üretim hızından yüksek olduğu için bu işlem sistemlerin üretim hızlarına da önemli bir etkisi olmamaktadır.

2.4. Ses Verileri Yardımıyla Rastgele Sayı Üreteci Tasarımı

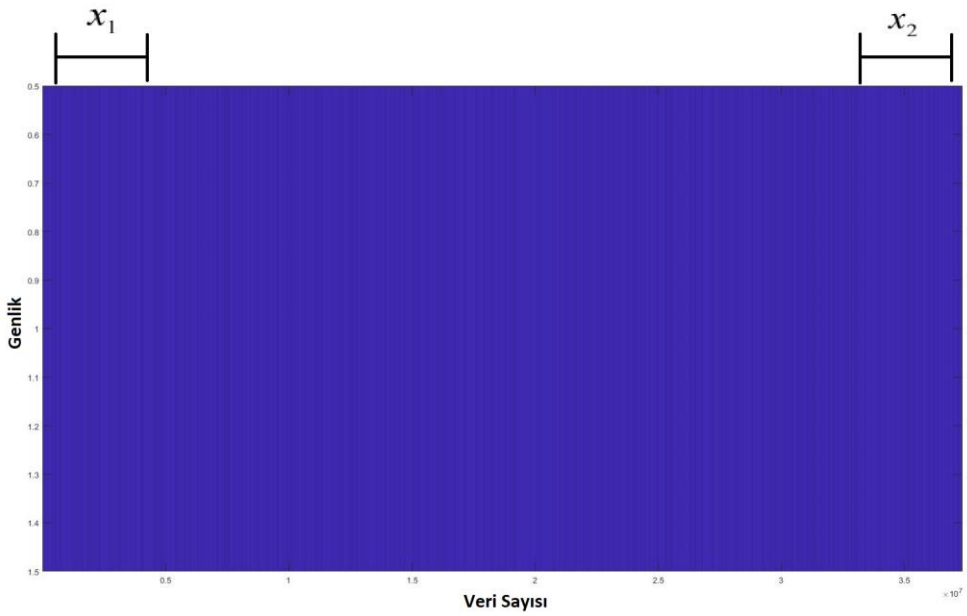
Ses verileri rastgele sayı üreteci tasarımları içerisinde kullanılmaya uygun yapılardandır. Nitekim literatürde ses verileriyle RSÜ tasarımı yapan çalışmalar mevcuttur [145–147]. Ses verisi çalışmalar içerisinde görüntü kadar yer tutmasa da en önemli iletişim kaynağı olarak öne çıkmaktadır. Hemen her ortamda ses verisi elde etmek mümkün olduğu gibi hazır ses veri setleri de mevcuttur.

Yapılan tez çalışmasında öncelikle kendi kaydettiğimiz ortam seslerini daha sonra da diğer çalışmalarla oluşturulmuş veri setlerini kullanarak GRSÜ tasarımları yapılmıştır. Yaklaşık bir saatlik ortam sesi kaydından sonra elde edilen veri seti Şekil 2.6'da görülmektedir.



Şekil 2.6. Bir saatlik ortam sesi kaydı

Kaydedilen yaklaşık bir saatlik ortam sesleri MATLAB platformu üzerinde kolayca ikili sayı formatına dönüştürülerek ham rastgele veri seti oluşturulabilir. Kaydedilen sesler ikili sayı formatında Şekil 2.7'de görünmektedir.



Şekil 2.7. İkili sayı formundaki kaydedilen ortam sesi verisi

Kaydedilen seslerden şekilde gösterildiği gibi x_1, x_2 olarak veri kümeleri seçilerek RSÜ tasarımları içerisinde kullanılabilir. Burada 1 saatlik ses kaydından yaklaşık olarak 86 milyon bit üretimi gerçekleştirilebilmektedir.

Mono formatta alınan ses kaydı 48 kHz örnekleme frekansıyla kaydedilmiştir. Ayrıca maksimum ses genliği 0.8 olarak belirlenmiştir. Kendi yaptığımız ses kaydı dışında farklı ses veri setleri de GRSÜ tasarımları içerisinde değerlendirilerek karşılaştırma yapılmıştır.

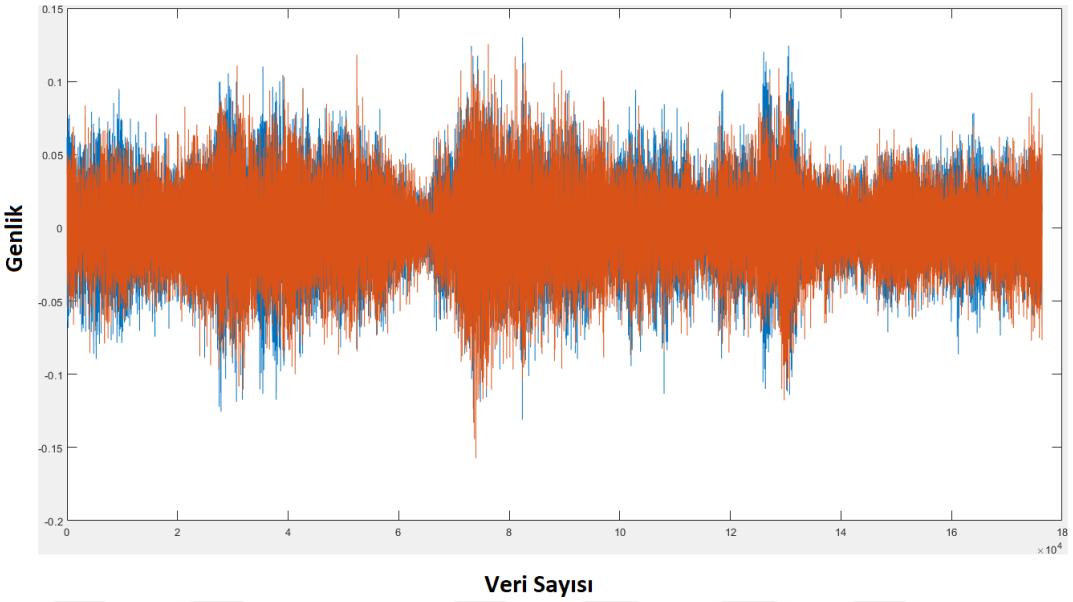
2.5. UrbanSound8K Ses Veri Seti

UrbanSound8K veri seti, kategorize edilmiş farklı ortam seslerinden oluşan bir ses kümesidir. Toplamda 8732 adet etiketlenmiş ses verisi içerir. 10 farklı kategoriye bölünen ses verileri 0 ile 9 arasında numaralandırılmıştır. Tablo 2.1'de kategorize edilmiş olan ses verilerinin etiket numarası karşılıkları gösterilmiştir [148].

Tablo 2.1. UrbanSound8K ses etiketleri

Etiket Numarası	Ses Türü
0	Havalandırma Tesisatı Sesi
1	Araba Kornası
2	Çocuk Sesleri
3	Köpek Havlaması
4	Delme İşlemi Sesi
5	Araç Motoru Sesi
6	Silah Sesi
7	Çekiç Sesi
8	Siren Sesi
9	Sokak Müziği

Bu şekilde sınıflandırılmış olan ses verileri için temel olarak yapay zeka ile kategorize etme uygulamaları yapılması düşünülmüştür [149]. Bu veri setinde sunulan ses kayıtlarının tamamı 4 saniyenin altındaki kısa ses kayıtlarıdır. Kaydedilen seslerin örnekleme frekansı 44.1 kHz olarak belirlenmiştir. Ayrıca tüm seslerin genlik değeri maksimum 0.65 olarak tespit edilmiştir. Örnek bir ses dosyası Şekil 2.8'de verilmiştir.

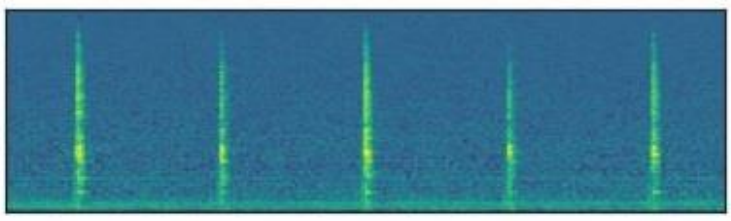


Şekil 2.8. UrbanSound8K örnek ses verisi

Bu veri setindeki sesler mono değil de stereo olarak yani çift kanallı olarak kaydedilmiştir. Bu da her bir kanaldaki ses için ayrı ayrı GRSÜ tasarımı yapılabileceğini göstermektedir.

2.6. ESC-50 Ses Veri Seti

ESC-50 veri seti Piczak tarafından 2015 yılında literatüre sunulmuştur [150]. Toplamda 2000 adet ses verisi içeren sette 5 saniye uzunluğundaki ses kayıtları bulunmaktadır. 44.1 kHz örnekleme hızıyla oluşturulan veri setinde 50 farklı ses sınıflandırması bulunmaktadır. Örnek bir ses kaydı Şekil 2.9'da gösterilmiştir.



Şekil 2.9. ESC-50 ses verisi örneği

Bu veri setinde bulunan seslerin sınıflandırılmış hali Tablo 2.2'de verilmiştir. 5 ana sınıf içerisinde 10 farklı ses türü gösterilmiştir.

Tablo 2.2. ESC-50 veri seti ses etiketleri

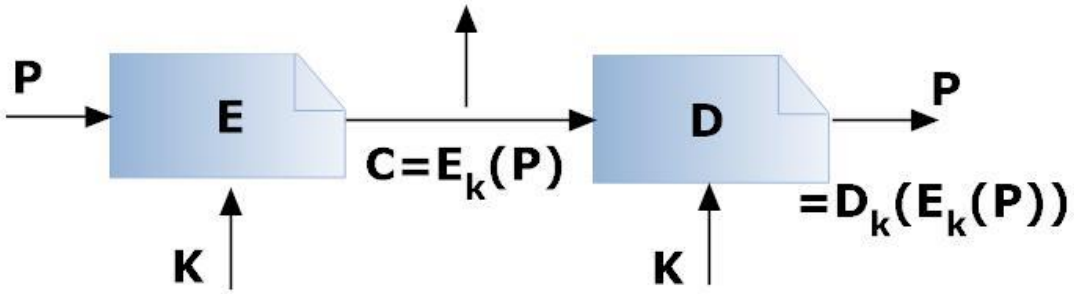
Hayvan Sesleri	Doğal Sesler ve Su Sesleri	İnsan Sesleri	İç Mekan Sesleri	Dış Ortam Sesleri
Köpek	Yağmur	Bebek Ağlaması	Kapı Çalma	Helikopter
Horoz	Deniz Dalgası	Hapşırma	Fare Tıklaması	Testere
Domuz	Odun Ateşi	Alkış Çalma	Klavye Sesi	Siren
İnek	Cırcır Böceği	Nefes Alma	Kapı Gıcırtısı	Korna
Kurbağa	Uçan Kuşlar	Öksürük	Şişe Açma	Motor
Kedi	Su Damlası	Ayak Sesi	Çamaşır Makinesi	Tren
Tavuk	Rüzgâr	Gülmek	Süpürge	Zil
Uçan Böcekler	Su Akışı	Diş Fırçalama	Saat Alarmı	Uçak
Koyun	Tuvalet Sifonu	Horlama	Saat Sesi	Havai Fişek
Karga	Şimşek	Şapırdatmak	Cam Kırma	Bıçkı

Toplam 50 farklı sınıflandırma bulunan ESC-50 veri setinde her bir kategoride 40 örnek bulunmaktadır. Farklı versiyonlarında 10 ve 20 sınıflandırma grubu olan veri setleri de bulunan ESC ses veri grubu genellikle yapay zeka ile ses türlerini sınıflandırma uygulamalarında kullanılmaktadır.

3. ŞİFRELEME TEKNİKLERİ

Günümüzde her türlü veri iletiminin gerçekleştirilebilmesi için çeşitli şifreleme sistemleri kullanılmaktadır. Özellikle kişisel verilerin korunması amacıyla şirketler bu konuda önemli yatırımlar yapmaktadır. Artık yasal bir yükümlülük haline de gelen kişisel verilerin korunması farklı amaçlarla gerçek kişilerin verilerinin açık şekilde paylaşılmasını ya da kötü amaçla kullanılmasını engellemektedir.

Şifreleme işlemleri, genellikle matematiksel veya mantıksal bir takım operasyonlara dayanır. Şifreleme algoritmasının genel yapısını tanımlayan bu işlemler genellikle herkese açık şekilde sunulur ve isteyen herkes bu verilere erişebilir. Ancak şifreleme işlemlerinde kullanılan şifreleme anahtarı sadece istenilen kişilerle paylaşılır. Şifreleme işlemlerinde, Şekil 3.1'de verilen yapı yaygın olarak kullanılmaktadır. Orijinal veri (P) alınarak şifreleme anahtarı (K) ile şifreleme algoritması (E) işlemine sokularak şifrelenmiş veri (C) elde edilir. Eğer kullanılan şifreleme algoritması simetrik anahtar kullanıyorsa aynı şifreleme anahtarı (K), şifrelenmiş veri (C) ile birlikte şifreleme işleminin tam tersini gerçekleştiren şifre çözme algoritmasıyla (D) işleme sokularak orijinal veri tekrar elde edilmiş olur [27].



Şekil 3.1. Şifreleme ve şifre çözme uygulamalarındaki genel işlem yapısı [1]

Günümüzde haberleşme ortamları yoğunlukla internet ve kablosuz haberleşme sistemleri olduğu için iletilen veriye ulaşmak zor değildir. Haberleşme kanalları kolay bir şekilde dinlenilerek veri elde edilebilir. İletilmek istenen veri, şifrelenerek başkaları tarafından bu veriler ele geçirildiğinde kullanılamaz hale gelmesi hedeflenmektedir. Bu işlem yapılırken temel amaç, istenmeyen kişilerin verilere ulaşamaması ve istenilen kişinin de kolayca istenen veriyi elde etmesini sağlamak olmalıdır [50].

Şifreleme sistemleri, temel olarak şifreleme anahtarının kullanım durumuna göre simetrik ve asimetrik anahtarlı şifreleme sistemleri olarak ikiye ayrılırlar. Her iki sistemde de kullanılan

şifreleme anahtarının tamamı ya da bir bölümü gizli bir şekilde kullanıcılara sunulmak zorundadır [66].

Şifreleme anahtarının seçimi, her türlü kriptolojik sistemde en önemli parametrelerden birisidir. Bu seçimi yaparken, rastgele sayı üreteçlerinden faydalanmak en doğru yol olacaktır. Böylece anahtarın tahmin edilebilme ihtimali zayıf olacaktır. Bunun yanı sıra rastgele sayı üreteçleri istatistiksel uygulamalar gibi farklı alanlarda da kullanılmaktadır.

Entropi kaynakları GRSÜ tasarımının temelini oluşturmaktadır. Bu kaynak, üretilen sayılardaki rastgeleliğin temelini oluşturmaktadır. Bu açıdan, seçilecek olan kaynak tasarımda kullanılmadan önce test edilmesi faydalı olur.

Kaotik sistemlerin GRSÜ kaynağı olarak kullanımı, literatürde geniş yer tutmaktadır. Kaotik davranış sergilediği kanıtlanan bir yapının entropi kaynağı olarak ele alınması GRSÜ tasarımında büyük kolaylık sağlayacaktır.

Şifreleme araştırmaları günümüzde genellikle görüntü şifreleme üzerine yoğunlaşmıştır [151–154]. Görüntü iletiminin popülerleşmesinin yanı sıra, araştırmaların görsel olarak sunulabilir olması da bu durum da etkili olmuştur. Birçok şifreleme sistemi her türlü veri de kullanılabilir olmasına rağmen, görüntü şifreleme uygulamalarını araştırmalarında sunarak sistemin güvenilirliği kanıtlamaktadır. Görüntü şifreleme sistemleri için özel güvenlik testlerinin bulunması da yine bu araştırmaların yaygınlaşmasında etkili olmuştur. Tezin Materyal ve Metot bölümünde bu özel testler detaylı olarak anlatılmıştır.

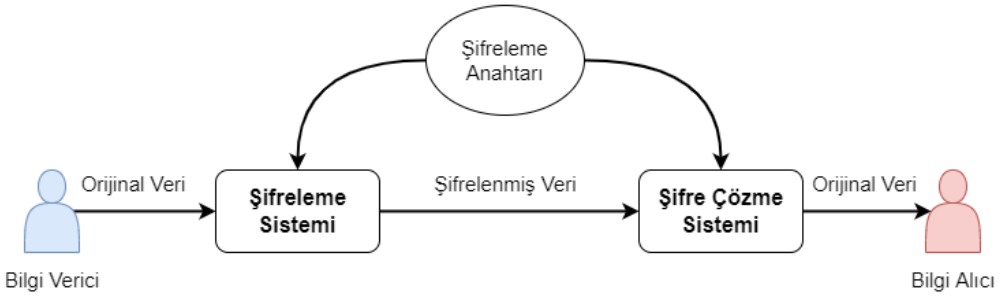
Şifreleme sistemleri, günümüzde temel olarak simetrik anahtar kullanan ve asimetrik anahtar kullanan şifreleme sistemleri olarak iki bölüme ayrılmaktadır. Her iki şifreleme sistemi de birbirine göre bazı avantaj ve dezavantajlara sahiptir [155–161]. Geliştirilen sistemler yeni bir şifreleme algoritması önerebildiği gibi mevcut standart olarak kullanılan DES, RSA ve AES gibi algoritmaların modifiye edilmesiyle de elde edilmiş olabilir [162–166]. Kaos tabanlı şifreleme sistemleri ise son yıllarda üzerine en çok çalışmanın yapıldığı araştırma alanı olmuştur [167–176]. Yapay zeka modellemeleriyle gerçekleştirilen bazı kriptolojik sistemler literatürde ilgi görmeye başlamıştır [177–179]. Yapılan diğer çalışmalarda, Ikbal ve Gopikakumari sıra tabanlı eşlenmiş gerçek dönüşüm kullanarak yeni bir renkli görüntü şifreleme sistemi tasarlamıştır [180]. Shi vd. yaptıkları çalışmada yeni bir sinyal örnekleme tekniği kullanarak gri tonlamalı görüntüler için şifreleme sistemi tasarlamıştır [181]. Shukla vd. yaptıkları çalışmada bulut sistemleri için bir şifreleme algoritması geliştirmiştir. Tasarlanan sistem MATLAB platformu üzerinde gerçekleştirilmiştir [182]. Raghuvanshi vd. yaptıkları çalışmada Brownian hareketlerini difüzyon yöntemi olarak kullanan yeni bir şifreleme sistemi tasarımını gerçekleştirmiştir [183]. Ma vd. yaptıkları çalışmada kaos tabanlı bir blok görüntü şifreleme algoritmasının kriptanalizi üzerine araştırmalar yapmıştır. Elde edilen sonuçlar ışığında seçilen metin saldırılarında sistemin dirençsiz olduğu belirlenmiştir [184]. Liu vd. yaptıkları çalışmada yeni bir görüntü şifreleme sistemi

önermiştir. Önerilen sistem S-kutuları yardımıyla görüntü şifreleme uygulamalarında denenmiştir [185].

Tüm bu çalışmalar neticesinde şifreleme sistemlerinin birbirinden çok farklı yapılarda olabileceği görülmektedir. Her yöntemin belirli avantaj ve dezavantajları olsa da en önemli ölçüt belirli güvenlik kriterlerinin sağlanmasıdır. Bu amaçla yapılan çalışmalarda en çok eleştirilen ve üzerinde çalışılan bölüm kriptanaliz yöntemleri olmuştur. Eğer geliştirilen sistemin kriptanaliz yöntemleri yardımıyla güvenilirliği kanıtlanırsa hangi yöntemden faydalanarak tasarlandığının önemi kalmamaktadır.

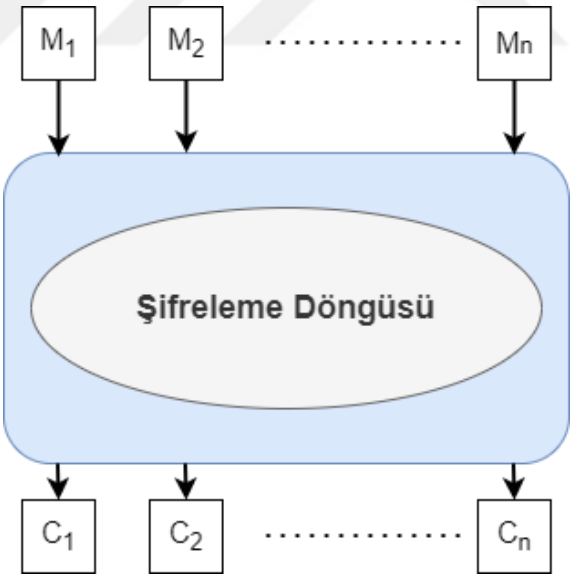
3.1. Simetrik Anahtarlı Şifreleme Yöntemi

Simetrik anahtarlı şifreleme sistemleri ile kriptolojik uygulamalar gerçekleştirilirken hem şifreleme hem de şifre çözme operasyonları için tamamen aynı anahtar kullanılır. Bu işlemler yapılırken algoritmada kullanılan anahtarın gizliliği esas alınmaktadır. Bu yöntem anahtarın gizli olmasını gerektirdiğinden gizli anahtar şifreleme yöntemi olarak da bilinir. Simetrik anahtarlı kriptolojik sistem için genel blok diyagram Şekil 3.2'de gösterilmiştir. Şekilde görüldüğü gibi, orijinal veri bir şifreleme algoritması ile şifrelendikten sonra iletildiği ortamda şifrelenmiş veri aynı anahtarı kullanan şifre çözme algoritması yardımıyla orijinal veriye döndürülmektedir. İki işlem içinde aynı anahtarın kullanılması bu sistemin en büyük özelliğidir. Şifreli verinin başarılı bir şekilde çözülebilmesi için ortak kullanılan anahtarın mutlaka bilinmesi ve üçüncü şahıslardan gizlenmesi bir zorunluluktur. Günümüzde yaygın bir şekilde kullanılan simetrik şifreleme teknikleri, genellikle asimetrik şifreleme tekniklerine göre daha hızlı çalışırlar. Bu sistemleri donanımsal olarak devreye almak kolaydır. Ancak şifreli veriye yapılabilecek olası şifre kırma saldırılarına karşı dayanıklılığı azdır. Anahtarın veri iletilmek istenen kişiye önceden güvenli bir şekilde iletilmesi de farklı bir dezavantaj oluşturmaktadır. DES, AES, TEA, Blowfish, SEA, IDEA ve RC5 gibi algoritmalar bu şifreleme sistemlerine en iyi örneklerdir [186]. Bu algoritmaların yapısında şifrelenecek olan veriyi bloklara bölmek vardır. Bloklara bölünen veri ayrı ayrı şifrelenerek algoritma tarafından belirlenen blok uzunluklarında sabit veriler üretilir. Şifrelenmiş olan veriler çözülürken de yine bloklar halinde işlem yapılır ve orijinal veri elde edilmiş olur. Bu şifreleme yöntemi de diğer şifreleme yöntemleri gibi elde edilecek olan veriler üzerinde en yüksek oranda karmaşıklık sağlamayı hedefler.



Şekil 3.2. Gizli anahtar ile şifreleme yöntemi diyagramı

Örnek olarak, Şekil 3.3'de gösterilen blok şifreleme yapısı ele alınabilir. M olarak tanımlanan mesaj verileri şifreleme işlemine tabi tutulduktan sonra C şifrelenmiş metinleri elde edilir. Her bir M mesaj metni ve C şifrelenmiş metinin blok boyutu birbirine eşittir. M_n şeklinde tanımlanan mesaj metni sayısı paralel olarak aynı sayıda C_n şifrelenmiş metnini üretecektir. Blok şifreleme sistemlerinin en büyük avantajı mesaj blokları hazırlandıktan sonra paralel olarak da şifreleme işlemlerinin yapılabilmesidir. Her blok birbirinden bağımsız olduğu için şifreleme işlemlerinin belirli bir sırayla yapılması gerekmez. Bu da günümüzdeki iletişim sistemlerinin hızlı çalışması için en büyük avantajdır.



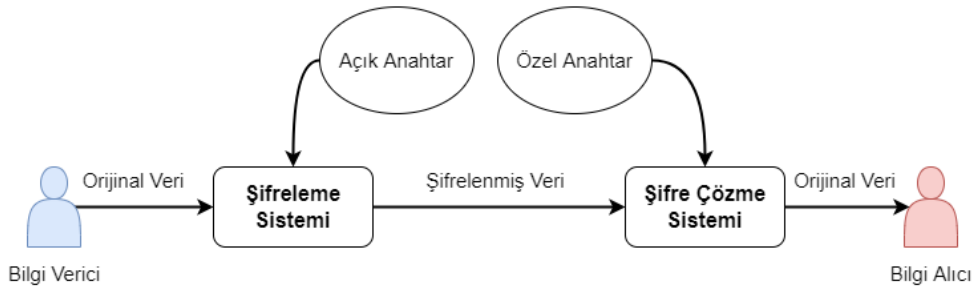
Şekil 3.3. Blok şifreleme sistemlerinin genel işlem sıralaması

RC4, A5/1, A5/2, Panama algoritması gibi algoritmalar ise akış şifreleme algoritmaları olarak bilinmektedir. Bu sistemler dizi şifreleme algoritmaları olarak da kullanılmaktadır. Genellikle kullanılan algoritmaların özelinde şifreleme işlemleri bit ya da bayt yapısında sıralı olarak yapıldığı için her bir işlem öncelikle bağlantılı olmaktadır. Böyle bir bağın bulunması

şifreleme sisteminin herhangi bir adımında hatayla karşılaşılması durumunda tüm verinin yok olmasına neden olabilir. Bu olay şifreleme algoritmasının hassasiyeti olarak tanımlanan bir parametreye bağlıdır. Şifre çözme işlemi uygulanırken de benzer şekilde en ufak hata durumunda tüm işlem yanlış sonuçlanarak orijinal verinin elde edilmesi imkânsızlaşabilir.

3.2. Asimetrik Anahtarlı Şifreleme Yöntemi

Diğer bir adı açık anahtar şifreleme tekniğidir. Şifreleme ve şifre çözme işlemlerinde kullanılan anahtar açık bir şekilde verilmektedir. Ayrıca asimetrik anahtarlı şifreleme için özel bir anahtar türü de bulunmaktadır. Şekil 3.4'te görüldüğü üzere şifreleme işlemi için kullanılan anahtar açık anahtar, şifre çözme için kullanılan anahtar ise özel anahtar olarak tanımlanmaktadır. Açık anahtar ile eldeki veriyi sadece şifreleme işlemi yapılabilir. Özel anahtara sahip kişiler ise şifreyi çözerek orijinal veriyi elde edebilir. Şifre çözme işlemi için kişiye özel anahtar bulunması şifreleme işleminde kullanılan açık anahtarın gizliliğini gerektirmemektedir. Bu anahtarın herkes tarafından bilinmesi veri güvenliğini herhangi bir şekilde etkilemezken şifreleme anahtarını bilen bir kişinin şifre çözme işlemi yapması mümkün değildir.



Şekil 3.4. Asimetrik şifreleme yöntemi

Asimetrik anahtarla şifreleme yönteminde her kullanıcıya ait farklı bir anahtar mevcuttur. Böylece birçok kullanıcı arasından istenilen kullanıcıya ayrı mesaj gönderilebilir. Bu özellikleri sebebiyle asimetrik anahtarlı şifreleme yöntemlerinin simetrik anahtarlı şifreleme yöntemlerine göre saldırılara daha dayanıklı olduğu söylenebilir. Ancak şifreleme hızları karşılaştırıldığında simetrik anahtarlı şifreleme tekniğinin büyük bir üstünlüğü mevcuttur. Simetrik anahtarlı şifreleme tekniği bütünlük, anahtar yönetimi, gizlilik, kimlik denetimi olarak ortaya çıkan problemleri asimetrik anahtarlı şifreleme tekniği ile çözebilir.

Şifreleme anahtarının uzunluğu şifreleme güvenliğiyle yakından ilgilidir. Genel itibariyle şifreleme anahtarı ne kadar uzunsa sistem o kadar güvenlidir denilebilir. Ancak özellikle bazı asimetrik şifreleme algoritmaları bazı anahtar uzunluklarını kullanmak için uygun değildir. Simetrik şifreleme algoritmalarında ise her türlü uzunluktaki anahtar kullanımı daha uygundur.

Ayrıca şifrelenecek veri tipine göre anahtar uzunluğunun belirlenmesi haberleşme sistemindeki güvenliği daha fazla arttıracaktır.

Şifreleme uygulamalarında kullanılan anahtarların rastgele olması en önemli kriterlerden biridir. Günümüzde bu ihtiyacı gidermek adına bazı çalışmalar yapılmıştır. İstatistiksel testleri başarıyla geçmiş bir rastgele sayı üreteci tasarımı şifreleme anahtarı olarak kullanıldığında yüksek güvenlik sağlanmış olur.

4. MATERYAL VE METOT

Rastgele Sayı Üreteçleri tasarımı için birçok farklı yöntem bulunmaktadır. Halka osilatörle yapılan çalışmalar GRSÜ sistemleri içerisinde en yaygın kullanılan yöntem olarak öne çıkmaktadır [135]. Farklı yöntemlerle binlerce çeşit rastgele sayı üreteci tasarlamak mümkündür. Önemli olan bir entropi kaynağı oluşturmaktır. Oluşturulan entropi kaynağını tasarımlar içerisinde kullanmak için belirli düzenlemeler yapmak gerekebilir. Örnek olarak literatürde sıklıkla rastlanan gürültü sinyali temelli GRSÜ'leri ele alabiliriz [187]. Alınan sinyalin 0.1 ile 1 Volt arasındaki gerilim sinyalleri olduğunu varsayarsak ve üretilen rastgele sayıların bit düzeyinde olmasını bekliyorsak, yapılması gereken işlem öncelikle elde edilen ölçüm değerlerinin normalize edilmesidir. Yani bit düzeyinde işlem yapabilmek için sinyalin öncelikle sayısallaştırılması gerekir. Ardından sinyal incelenerek Lojik-1 ve Lojik-0 değerlerini temsil edecek ölçüm değerleri belirlenir. Bu aşama normalizasyon olarak adlandırılabilir. Elde edilen değerler doğrudan RSÜ çıktısı olarak kullanılacağı gibi son-işlem algoritmalarıyla istatistiksel özellikleri geliştirilebilir. Bu aşamada, giren istatistiksel analizlerle tasarlanan RSÜ'nün kalitesi ortaya konulabilir. Ayrıca, son-işlem algoritmalarının kullanılmasına gerek olup olmadığı da bu şekilde belirlenebilir. İstatistiksel özellikleri iyi olan ham bir RSÜ son-işlem algoritmaları kullanılmadan da gerçekleştirilebilir. Ancak istatistiksel olarak iyi sonuçlar alamayan RSÜ tasarımları son-işlem algoritmalarına ihtiyaç duyar.

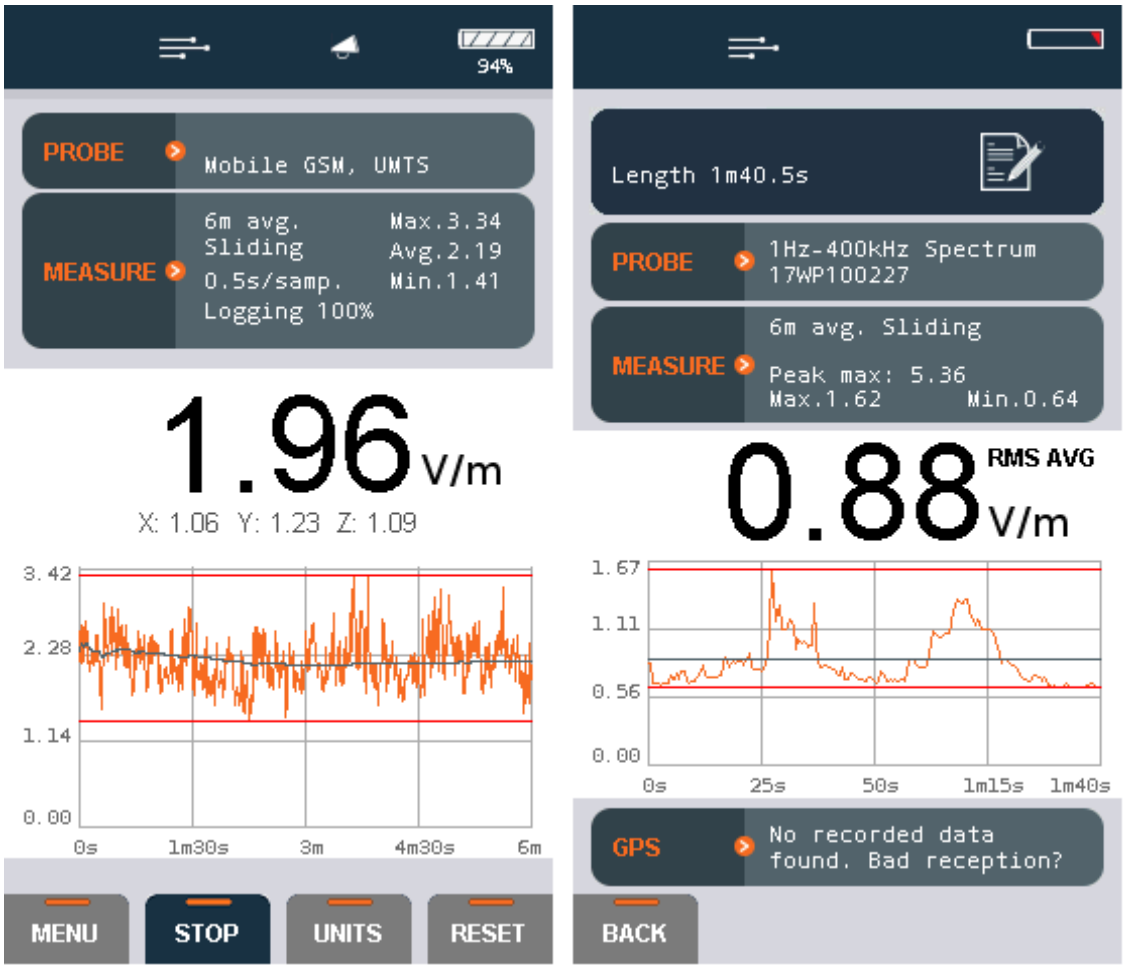
Tezin bu kısmında öncelikle çalışmada kullanılan rastgele sayı üreteci yöntemlerinden bahsedilecektir. Ardından bu rastgele sayı üreteçlerine ilişkin istatistiksel analizlere değinilecektir. RSÜ tasarımlarına ilişkin son-işlem algoritmaları hakkında detaylı bilgiler sunulacaktır. Materyal ve Metot kısmının sonunda şifreleme işlemleri hakkında bilgiler ve şifreleme sistemlerinde kriptanaliz yöntemlerinden bahsedilerek bu bölüm tamamlanacaktır.

4.1. GRSÜ Tasarım Yöntemi

Tez çalışması içerisinde çok farklı rastgelelik kaynağıyla çalışabilecek bir GRSÜ sistemi tasarlanmıştır. Burada yapılacak olan işlem öncelikle entropi kaynağı verinin hazırlanmasıyla başlamaktadır. Normalize edilen veriler doğrudan GRSÜ sistemi içerisinde kullanılabilir. Eğer istatistiksel olarak problemler varsa GRSÜ'yü geliştirmek için son-işlem algoritmaları uygulanır. Burada en önemli parametrelerden birisi de ihtiyaç duyulan RSÜ'nün teknik özellikleridir. Üretilen bitlerin hızı, toplam bit sayısı, enerji tüketimi gibi parametrelere göre uygun RSÜ tasarımı gerçekleştirilmelidir.

Tez çalışmasında kullanılan ve sayı üreteci kaynaklarından biri olan elektriksel sinyaller için tasarlanan GRSÜ modelinin matematiksel işlem adımları bu başlıkta verilmiştir. $x(m)$ sinyalleri

volt/metre olarak ölçülen elektrik alan değerleri olduğunu varsayarsak mevcut veri setinde 0,05 ile 5,81 V/m değer aralığında elektrik alan verileri bulunmaktadır. Burada herhangi bir ölçüm değeri kullanılabilmektedir. Örnek ölçüm değerleri Şekil 4.1'de gösterilmiştir.



Şekil 4.1. Örnek elektrik alan ölçümleri

Burada elde edilen sinyallerle GRSÜ tasarlayabilmek için veriler çeşitli işlemlerle bit düzeyine indirgenmektedir. $x(m)$ sinyalleri denklemdeki gibi y_t değerine dönüştürülür. Burada t parametresi sayı basamağını temsil etmektedir.

$$y_t = x(m).10^5 \tag{4.1}$$

Daha sonra elde edilen bu değerler modüler aritmetik işlemleriyle z_t değerlerine dönüştürülür.

$$z_t = y_t \bmod 2 \quad Eğer \rightarrow t \in [1,5] \tag{4.2}$$

Burada t parametresinin [1,5] aralığında alınmasının temel sebebi alınan ölçüm değerlerinin 5 basamak içermesinden dolayıdır. Basamak sayılarının farklı olması durumda bu parametre isteğe göre değiştirilebilir.

$$k = \left(\sum_{t=1}^{5} z_t \right) \bmod 2 \tag{4.3}$$

z_t değerlerinin toplamının ikili modülasyonu alınarak, k ile simgelenen ham rastgele bit dizisi oluşturulmuştur. Bu bit dizisi doğrudan GRSÜ olarak kullanılacağı gibi son-işlem uygulanarak da kullanıma sunulabilir.

4.2. SRSÜ Tasarım Teknikleri

Tez çalışması içerisinde farklı SRSÜ tasarımları da gerçekleştirilmiştir. Burada gerçekleştirilen tasarımların genel özelliği hızlı bit üretimi sağlayan Rastgele Sayı Üreteçlerinin tasarlanmasıdır. Ayrıca burada geliştirilen SRSÜ'ler kriptolojik amaçlarla kullanılacağı için belirli özellikleri taşımaları gerekmektedir. Bir sonraki konuda anlatılacak olan istatistiksel değerlendirme yöntemlerine göre başarılı olan tasarımlar şifreleme sistemleri içerisinde değerlendirilebilecektir. İstatistiksel olarak zayıf tasarımlar son-işlem algoritmaları yardımıyla kullanılabilir hale gelse de bit üretim hızına dikkat etmek gerekir.

SRSÜ tasarımlarında aşağıdaki parametreler en önemli etmenlerdir:

- SRSÜ'ler algoritma tabanlı olduğu için bit üretim hızı yüksek olmalıdır.
- Tasarlanan sistemin en basit mikroişlemcilerden, karmaşık yapılara kadar uygulanabilirliği önemlidir.
- Tasarlanan sistemler mobil uygulamalar da olabileceği için güç tüketimini minimize etmek önemli bir parametredir.
- Donanım ihtiyacını minimumda tutan sistemler (örneğin sadece temel lojik elemanlarla çalışan) uygulama açısından avantajlıdır.
- İhtiyaca göre gerekli rastgele bit uzunluğunun sağlanması gerekir.
- Tasarımlar istatistiksel olarak başarılı olmalıdır.
- İhtiyaca uygun olarak eğer gerekli ise sistemler aynı başlangıç parametreleriyle aynı sonucu vermelidir.

Buna göre tez çalışması kapsamında kullanılacak olan SRSÜ algoritmaları başlıklar halinde verilmiştir. Bu tasarımların her biri farklı ihtiyaçlara hitap etmesi sebebiyle farklı kullanım alanları vardır. Şifreleme işlemlerinde bu SRSÜ yöntemlerinin ne amaçla kullanılacağı detaylı olarak anlatılacaktır.

4.2.1. Doğrusal Eşlenik Üreteci Uygulaması

Doğrusal Eşlenik Üreteci (Linear Congruential Generator) algoritması en temel matematiksel işlemleri içermesi sebebiyle geniş bir kullanım alanı bulmaktadır. Temel bir algoritma olması ve uzun yıllardır kullanılması sebebiyle oldukça hızlı bir bit üretimi sağladığı söylenebilir. Sistem gereksinimleri de oldukça düşüktür. Her türlü işlemcide uygulamak mümkün olabilir.

Temel olarak bit üretim eşitliği denklemde verilmiştir.

$$X_i = [(A \times X_i) + B] \bmod (C) \tag{4.4}$$

Burada bit üretiminin başlayabilmesi için öncelikle bir X_i başlangıç değerine ihtiyaç duyulmaktadır. C değeri üretilmesi hedeflenen maksimum bit uzunluğunu göstermektedir. A ve B değerlerinin de seçilmesiyle farklı rastgele sayı dizileri oluşturmak mümkündür [29]. Oluşturulan sayı dizileri aynı başlangıç şartları için her zaman aynıdır. Şekil 4.2'de Python üzerinde bir Doğrusal Eşlenik Üreteci tasarlamak için gerekli kodlar verilmiştir.

```
def lcg(moddeğeri, a, b, başlangıçdeğeri):
    while True:
        seed = (a * başlangıçdeğeri + b) % moddeğeri
        yield başlangıçdeğeri
```

Şekil 4.2. Doğrusal eşlenik üreteci Python kodları

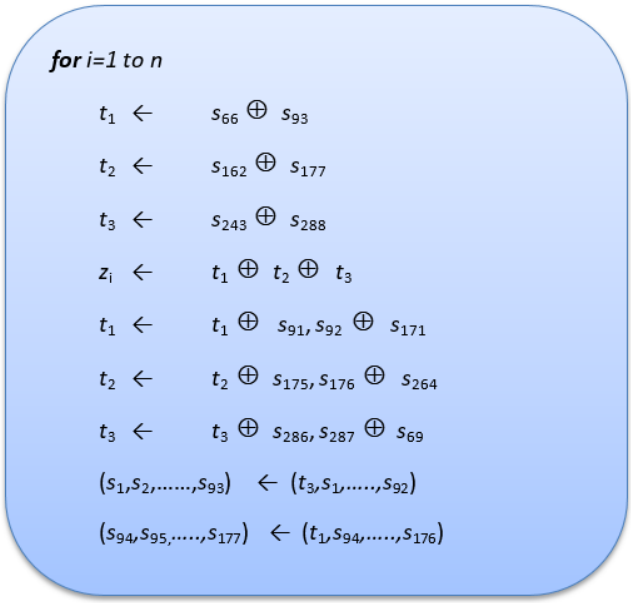
Burada LCG, Doğrusal Eşlenik Üreteci'ni temsil etmektedir. LCG yaygın bir algoritma olduğu için hemen her programlama dilinde gerçekleştirilebilir. Özellikle düşük sayı üretimi kapasiteli uygulamalarında belirli güvenlik zaafları oluşabileceği yapılan araştırmalar neticesinde kanıtlanmıştır [56]. Güncel ve sıklıkla bir programlama dili olması sebebiyle bu uygulama için Python tercih edilmiştir. Aynı zamanda Matlab üzerinde de tasarım hayata geçirilmiştir.

4.2.2. Trivium Algoritması

Trivium algoritması yine yüksek bit üretim hızlarına sahip olan bir algoritmadır. Bu algoritmanın en önemli yanı ise yüksek giriş parametresine ihtiyaç duymasıdır. Fazla başlangıç değerine ihtiyaç duyması sebebiyle genellikle ayrı bir SRSÜ olarak değil de son-işlem algoritması

olarak kullanımı daha yaygındır. Bu uygulamanın kriptolojik çalışmalarda kullanılabileceği araştırmalarla kanıtlanmıştır [25].

Trivium algoritmasının iç durum değişkenleri $s_1, s_2, s_3, \ldots, s_{288}$ ile gösterilmektedir. Buna göre bu algoritmanın akış şeması Şekil 4.3'te gösterilmiştir.

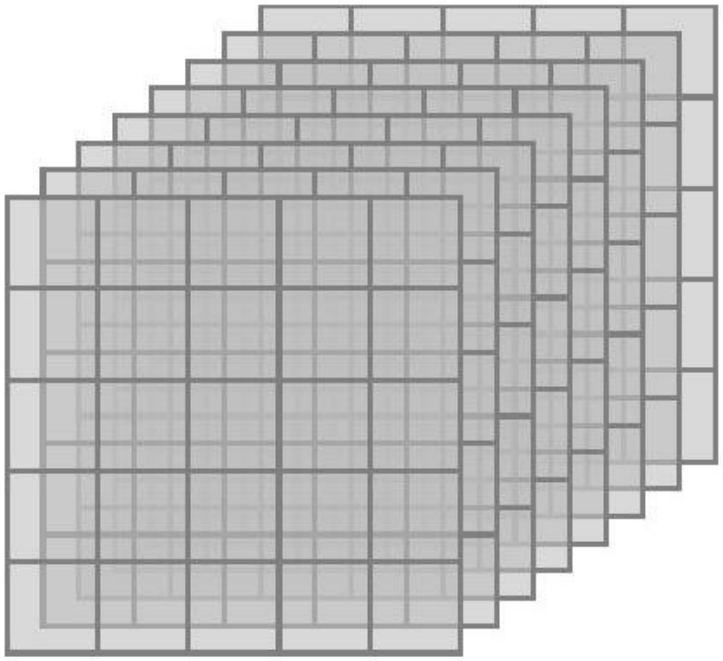


Şekil 4.3. Trivium algoritmasının işleyiş şeması [188]

Trivium algoritması bit düzeyinde senkron akış şifreleme yapan hızlı ve güvenli bir algoritma olarak da kullanılabilir. Enerji tüketimi de temel lojik elemanları kullanması sebebiyle oldukça düşüktür [188].

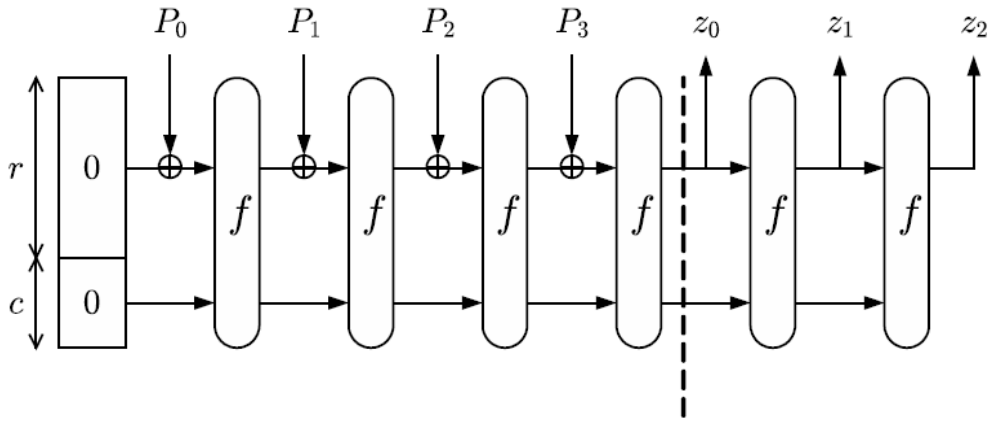
4.2.3. Keccak Algoritması

Keccak algoritması NIST tarafından özet algoritması olarak kullanılan bu alandaki en önemli algoritmalardan birisidir. Guido Bertoni tarafından ilk kez 2009 yılında temelleri oluşturulsa da yıllar içerisinde farklı araştırmacıların da katkılarıyla kullanımı yaygınlaştırılmıştır [189]. Bu algoritma Şekil 4.4'teki gibi sünger yapıları kullanmaktadır.



Şekil 4.4. Keccak algoritması sünger yapısı

Kriptografik özet algoritması olarak kullanılan Keccak ailesi algoritmaları r ve C olarak gösterilen giriş ve ara vektör isimli iki parametreden oluşur. r işlenen özet algoritma bloğunu, C ise güvenlik parametrelerinden oluşan bloğu oluşturmaktadır. f fonksiyonu mesaj bloklarını sıra sıra işleme alarak istenen özet değerini üretir. Bu işlemler Şekil 4.5'te gösterilmiştir.



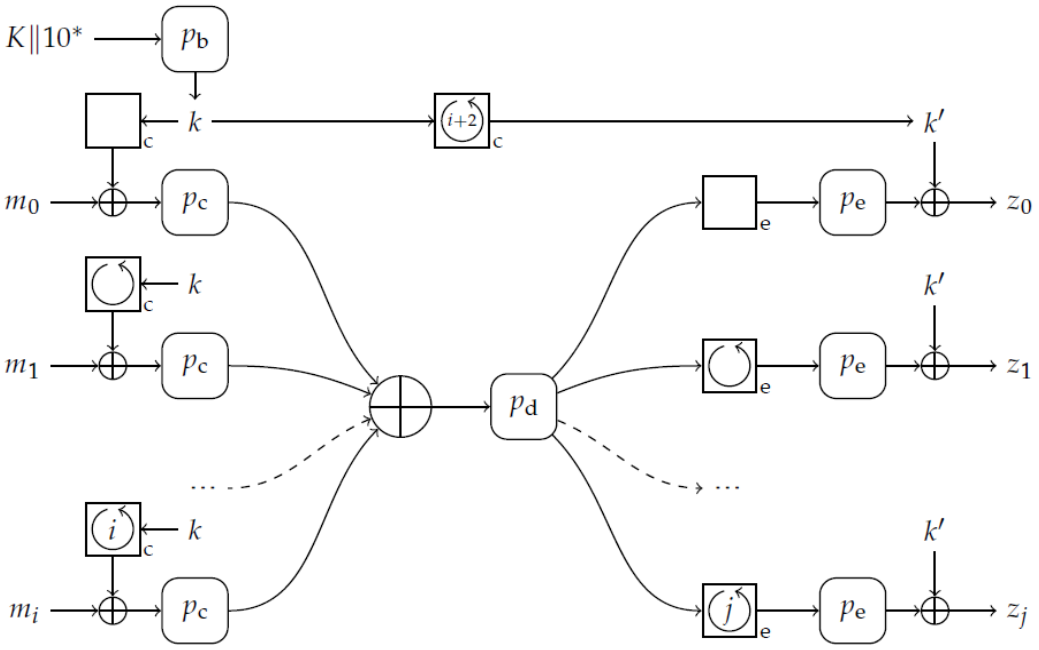
Şekil 4.5. Keccak algoritması yayılma ve absorbasyon adımları[189]

İlk blok için başlangıç değerleri 0 olmaktadır. Daha sonra üretilen her değer bir sonraki bloğun başlangıç değerini oluşturur. f fonksiyonu Keccak sünger yapısının temelini oluşturmaktadır.

Sünger yapıdaki fonksiyonlar ya da özet algoritmaları kimlik doğrulama işlemleri, akış şifreleme algoritmaları, blok şifreleme algoritmaları, rastgele sayı üreteçleri gibi farklı farklı kullanım alanlarına sahiptir. Özet algoritma standardı olarak SHA-3'de 2013 yılından itibaren güvenilir olarak kullanılmaktadır [190].

4.2.4. Farfalle Fonksiyonu Algoritması

Farfalle Fonkisyonu Keccak algoritma ailesi içerisinde yer alan bir fonksiyondur. Ufak farklılıklar içermesine rağmen temelde yine sünger fonksiyon yapılarını kullanarak işlem yapar. Farfalle fonksiyonu farklı uzunluktaki gizli anahtarları ve veri dizilerini alarak istenilen uzunluktaki bir özet değerine dönüştürmeyi hedefler. Genel yapısı Şekil 4.6'da gösterilen şekildedir.



Şekil 4.6. Farfalle fonksiyonu genel yapısı[191]

Farfalle fonksiyonunun güvenilirliği yapılan çalışmalarla kanıtlanmıştır [192]. Özet fonksiyon olarak oldukça başarılı olan Farfalle uygulama esnasında çeşitli kullanım modları sunmaktadır. Permutasyon tabanlı bir şifreleme algoritması olarak başarılı sonuçlar elde edebilir [191]. Tez çalışması içerisinde Farfalle fonksiyonu rastgele sayı üreteci tasarımları içerisinde kullanılmıştır.

4.3. Rastgele Sayı Üreteçlerinin Değerlendirilmesi

Rastgele sayı üreteçlerinin çıktılarının değerlendirilmesi için birçok yöntem bulunmaktadır. Genellikle istatistiksel olarak değerlendirme sunan bu yöntemler RSÜ kullanım alanlarına göre sınıflandırma işleminde de kullanılabilir.

En yaygın kullanılan test Amerikan Standartlar ve Teknoloji Enstitüsü tarafından yayınlanan NIST istatistiksel rastgelelik testleridir. 2010 yılında Rukhin vd. tarafından oluşturulan NIST 800-22 test aracı günümüzde halen kullanılmakta olan en güncel ve kapsamlı istatistiksel rastgelelik test aracıdır [193]. Buna benzer şekilde Diehard, FIPS-140, Crypt-X ve TestU01 gibi test araçları mevcuttur. Ayrıca otokorelasyon, entropi testi, scale-index testi gibi tekli testler de üretilen rastgele sayı dizilerinin farklı istatistiksel özelliklerini değerlendirme sürecinde faydalı olmaktadır.

Bu başlık altında tez çalışmaları kapsamında kullanılan istatistiksel testlerin detaylarından bahsedilecektir.

4.3.1. NIST-800-22 testi

NIST-800-22 istatistiksel rastgelelik testleri uluslararası kuruluşlarca onaylanan literatüre en çok kabul gören rastgele sayı üreteci değerlendirme yöntemidir. Toplam 15 test bulunan bu istatistiksel analiz testinde son iki test 1 milyon örnek gerektirdiğinden genellikle 1 milyonun üstünde örnek için test yapılmaktadır.

NIST-800-22 testleri toplamda 15 farklı test yardımıyla bit dizilerini değerlendirmek için yeterli görülse de farklı testlerin uygulanması güvenilirliği daha da arttıracaktır. NIST-800-22 testi çıktıları olarak farklı sayılarda P-değerleri elde edilmektedir. P-değerleri genellikle 0.001'den büyük olmalıdır, yani 0,001<P-değeri<1 şartını sağlayan istatistiksel testlerin başarılı olarak sonuçlandığı söylenebilir. P-değerleri zaten matematiksel olarak en yüksek "1" olacak şekilde formüle edilmiştir. P-değerleri genel kabul olan binde birin ya da bazı yerlerde yüzde birin altında hesaplandığında o test başarısızdır denilebilir.

Frekans testi, bit dizisi içerisindeki "1" ve"0" oranını incelemektedir. NIST-800-22 testleri içerisindeki frekans testi Denklem 4.5'de gösterilen şekilde P-değeri hesaplanarak değerlendirme yapılabilir.

$$P - değeri = erfc\left(\frac{S_{obs}}{\sqrt{2}}\right) \tag{4.5}$$

Bu denkleme göre hesaplama yapıldıktan sonra elde edilen P-değeri 0,001'den büyükse bit dizisinin testi geçtiği söylenebilir.

Bir blok içerisinde frekans testinde standart frekans testi değişik bir yöntemle uygulanır. Bit dizisinin blokları içerisinde "1" ve "0" dengesine bakılır. Bit dizileri daha önceden belirlenen M bit

boyutundaki bloklara bölünerek, bölünen her bir bloğun içerisinde bulunan "1" oranı değerlendirilir. Eğer M=1 olarak seçilirse teknik olarak bit dizisi bloklara ayrılamayacağı için test frekans testiyle aynı şekilde çalışır.

Akış testi, bir dizideki "1" ve "0" bloklarının değişim hızını inceler.

Bir blok içerisindeki en uzun akış testi, n bit uzunluğundaki bloklar içerisinde en uzun ardışık gelen"1" değerlerini inceler. P değeri Denklem 4.6'daki şekilde tanımlanmıştır.

$$P - değeri = igamc\left(\frac{K}{2}, \frac{\chi^2_{obs}}{\sqrt{2}}\right) \tag{4.6}$$

İkili matris derece testinde, bit dizisi düzenli bir şekilde matrissel forma dönüştürülür. Bu matrisin derecesinin hesaplanmasıyla matris blokları arasında doğrusal bir bağ olup olmadığı araştırılır. Testin istatistiksel değerlendirmesi için χ^2 dağılım hesabı kullanılır. Toplam oluşturulan matris sayısı $N=|n/MQ|$ şeklinde tanımlanmaktadır. Oluşturulan matrislerin dışında kalan bit'ler ihmal edilerek hesaplama yapılabilir.

Ayrık fourier dönüşüm testinde, bit dizisinin periyodik olup olmadığı araştırılır. Örnek bir bit serisi ile durum açıklanılmak istenirse; Bit dizisinde n=100 olarak seçilirse üzere ε = $\varepsilon_1,\varepsilon_2,\ldots\varepsilon_{100}$=1100100100001111110110101010001000100001011010001100001000110100110001001100011001100010100010111000 olarak belirlenebilir. Yapılacak dönüşüm sonucuna göre ε =1, 1,-1,-1, 1,-1,-1, 1,-1,-1 ,…, -1,-1,-1 şeklinde bir dizi elde edilmiş olur. Yani bit dizisinde "1" değerler yine "1", "0" olan değerler ise "-1" dönüştürülmüş olur. Elde edilen yeni ε dizisinde ayrık zamanlı Fourier dönüşümü uygulanarak S_j değeri hesaplanabilir. Bu hesaplama Denklem 4.7'de gösterilmiştir.

$$S_j = \sum_{k=1}^{n} ke\frac{(2\pi_i(k-1)j}{n} \tag{4.7}$$

Bu testte en önemli parametrelerden birisi tepe yüksekliğinin eşik değeri olarak gösterilen T = $\sqrt{3n}$ < %95 olacak şekilde hesaplama yapılmalıdır. Bu değer sağlanmadan hesaplama yapılırsa testin başarısız olması kaçınılmazdır.

Örtüşmeyen şablon eşleştirme testinde, bit dizisi içerisinde önceden belirlenen bir bit dizisinin bulunma sıklığı incelenir.

Örtüşen şablon eşleştirme testi ise farklı olarak örtüşmeleri bit kaydırarak devam ettirerek hesaplayarak testi tamamlar.

Evrensel istatistik testinde, bit dizisinin veri kaybı olmaksızın sıkıştırılıp sıkıştırılamayacağı incelenmektedir. Bu test ayrıca kriptolojik uygulamalarda gizli anahtarın kalite faktörünü belirlemekte kullanılır [194].

Doğrusal karmaşıklık testi, bit dizilerinin karmaşıklığını doğrusal geri beslemeli kaydırmalı kaydedici (Linear Feedback Shift Register) yardımıyla inceleyerek değerlendirme yapar.

Seri testi iki farklı test sonucuna ulaşarak dizinin rastgeleliğini inceler. Bu testte, dizi içindeki her m bitlik dizi örneğinin diğer m bitlik dizi örnekleriyle benzerliğini inceler.

Yaklaşık entropi testi, m bitlik dizi örneklerinin örtüşme frekansını inceler. Rastgeleliğin sağlanması için, ardışık ya da bitişik m bitlik blokların frekanslarını karşılaştırır.

Birikimli toplamlar testinin amacı bir dizinin tüm birikimli toplamlarının kısmi alt bloklarının birikimli toplamlarına oranını incelemektir.

Rastgele gezinimler testi, birikimli toplamların döngü sayısını tespit ederek "1", "0" değerlerini inceler. Diğer testlerden farklı olarak kısmi toplamlardan elde edilen sekiz farklı P değeri oluşturulur.

Rastgele gezinimler değişken testi ise birikimli toplamların rastgele yürüyüşte belirli durumların oluşma sayısını inceler. Bu testte de farklı olarak on sekiz P değeri hesaplanır.

Yukarıda bahsedilen testler NIST 800-22 test aracı içerisinde bulunan 15 farklı rastgelelik testinin detaylarıdır. Tablo 4.1'de NIST 800-22 test aracı içerisinde bulunan rastgelelik testleri verilmiştir.

Tablo 4.1. NIST 800-22 test çeşitleri

Test Numarası	Orijinal Test Adı	Türkçe Test Adı
1	Frequency Monobit Test	Frekans Testi
2	Frequency Test within a Block	Bir Blok içerisinde Frekans Testi
3	Runs Test	Akış Testi
4	Longest Run of Ones in a Block Test	Bir Blok içerisinde en Uzun Akış Testi
5	Binary Matrix Rank Test	İkili Matris Derece Testi
6	Discrete Fourier Transform Test	Ayrık Zamanlık Fourier Dönüşüm Testi
7	Non-Overlapping Template Matching Test	Örtüşmeyen Şablon Testi
8	Overlapping Template Matching Test	Örtüşen Şablon Testi
9	Universal Test (Maurer Test)	Evrensel Test (Maurer Testi)
10	Linear Complexity Test	Doğrusal Karmaşıklık Testi
11	Serial Test	Seri Testi
12	Approximate Entropy Test	Yaklaşık Entropi Testi
13	Cumulative Sums Test	Birikimli Toplamlar Testi
14	Random Excursions Test	Rastgele Gezinimler Testi
15	Random Excursions Variant Test	Rastgele Gezinimler Değişken Testi

Bir sayı dizisinin rastgele kabul edilebilmesi için NIST 800-22 içerisinde yer alan 15 testin tamamından başarılı olması gerekir. Tüm testlerden başarılı olamayan diziler bazı çalışmalara göre başarılı sayılabilseler de şüpheli bir durumun olduğu aşikardır. Ayrıca 14 ve 15 numaralı testlerin yapılabilmesi için minimum 1 milyon bitlik sayı dizisine ihtiyaç duyulduğu unutulmamalıdır [195]. RSÜ tasarımı yaparken bu parametreler göz önünde bulundurularak uygulama yapıldığında başarılı bir RSÜ ortaya koymak daha kolay olacaktır.

4.3.2. Autocorrelation Testi

Autocrrelation testi, rastgeleliği tespit etmek için kullanılan testlerden biridir. NIST testleri kadar yaygın olmasa da alınan sonuçlar önem arz etmektedir. Yapılan çalışmalarda NIST testlerini geçen serilerin Autocorrelation testinden kalabildiği belirlenmiştir [196].

Bu testin amacı, bit dizisi ile dizinin belirli oranlarda kaydırılmış versiyonları arasındaki ilişkiyi incelemektedir. Bu yolla dizinin rastgeleliğini tespit eder. s bit serisini belirtmek üzere, d dizideki kayma miktarını göstermektedir ve $1 \le d \le n/2$ şeklinde seçilir. Denklem 4.8'deki $A(d)$ değeri Autocorrelation değerini gösterecektir [197].

$$A(d) = \sum_{i=0}^{n-d-1} s_i \oplus s_{i+d} \tag{4.8}$$

Elde edilen $A(d)$ denklem 4.9'da yerine koyularak $|X5|$ ifadesi hesaplanarak Autocorrelation testi tamamlanır.

$$|X5| = 2\left(A(d) - \frac{n-d}{2} \right) / \sqrt{n-d} \tag{4.9}$$

Elde edilen $|X5|$ ifadesinin mutlak değeri 1.6449'dan küçük olması durumunda test başarılı olur.

4.3.3. Scale-Index Testi

Scale-Index testi, üretilen sinyallerin periyodik olup olmadığını anlamak için uygulanan bir testtir. Periyodikliği belirlemek için Sürekli Dalgacık Dönüşümünden faydalanır. Eşitliklerde dönüşüm işlemi gösterilmiştir [39].

$$Wf(u,s) = \left\langle f, \Psi_{u,s} \right\rangle = \int_{-\infty}^{+\infty} f(t) \Psi_{u,s}^{*}(t) dt \tag{4.10}$$

$$(s) = \left\| Wf(u,s) \right\| = \left(\int_{-\infty}^{+\infty} \left| Wf(u,s) \right|^2 du \right) \tag{4.11}$$

f Sürekli Dalgacık Dönüşümü enerjisini göstermek üzere, (s) skalayı göstermektedir.

$$S^{in}(s) = \left\| Wf(u,s) \right\|_{j(s)} = \left(\int_{c(s)}^{d(s)} \left| Wf(u,s) \right|^2 du \right)^2 \tag{4.12}$$

$J(s) = \left[c(s), d(s) \right] \subseteq I$ şartı sağlandığı zaman, normalize edilmiş S^{in} değeri denklem 4.13'de verilmiştir.

$$\overline{S}^{in}(s) = \frac{S^{in}(s)}{\left(d(s) - c(s) \right)^{1/2}} \tag{4.13}$$

Böylece Scale-Index testini uygulamak için gerekli değerler elde edilmiş olur ve denklem 4.14'deki oran test işlemi için kullanılmaktadır.

$$i_{scale} = \frac{S(s_{\min})}{S(s_{\max})} \tag{4.14}$$

İşlem sonucunda $0 \le i_{scale} \le 1$ aralığında bir değer elde edilir. Eğer bu değer 0'a yakınsa sistem periyodik, 1'e yakınsa sistem non-periyodik denilebilir.

4.3.4. Lyapunov Üstelleri Metoduyla Kaotik Davranış Tespiti

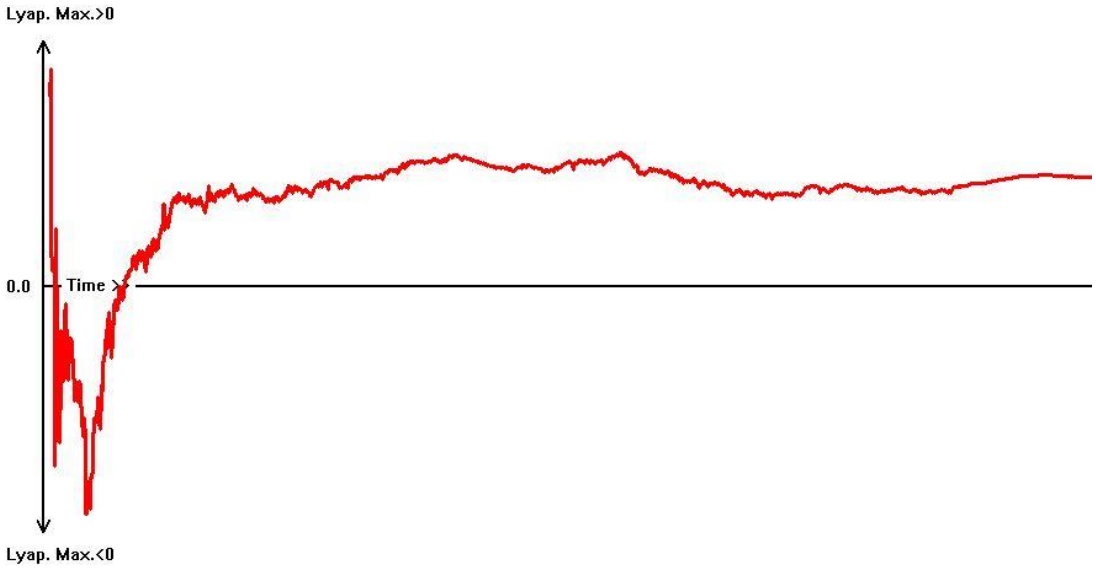
Lyapunov üstelleri non-lineer zaman serilerindeki kaotik davranışı belirlemek için yıllardır kullanılmaktadır. Çeşitli yollarla Lyapunov üstellerinden faydalanarak kaos tespit edilebilir. En basit haliyle en büyük dereceli Lyapunov üsteli 0'dan büyükse sistemin kaotik davranış sergilediği söylenebilir. En büyük Lyapunov üsteli denklem 4.15'deki şekilde hesaplanabilir [198].

$$\wedge_{\max} = \frac{1}{M.t_e} \sum_{i=0}^{M} \ln \frac{L_e^{(i)}}{L_0^{(i)}} \tag{4.15}$$

$\wedge_{\max}$ en büyük Lyapunov üsteli olmak üzere, L üsteller arasındaki Öklid mesafesini tanımlamaktadır. $\wedge_{\max}$ değeri 0'dan büyük olduğunda kaos tespit edilmiş olur.

Lyapunov üstelleri metodu genellikle entropi kaynaklarını analiz etmek için kullanılır. Doğrudan rastgele sayı üreteci analizlerinde kullanımı yaygın değildir. Kaotik bir devrenin ya da bir gürültü kaynağının rastgele davranış sergileyip sergilemediğini incelemek için kullanılması en

uygun yöntemdir [199]. Tez çalışmasında kullanılan elektromanyetik alan ölçüm değerlerinin ön analizini yapmak için Lyapunov üstellerinden faydalanılmıştır. Veri sayısına göre en büyük Lyapunov üstelinin değişimi Şekil 4.7'de verilmiştir.

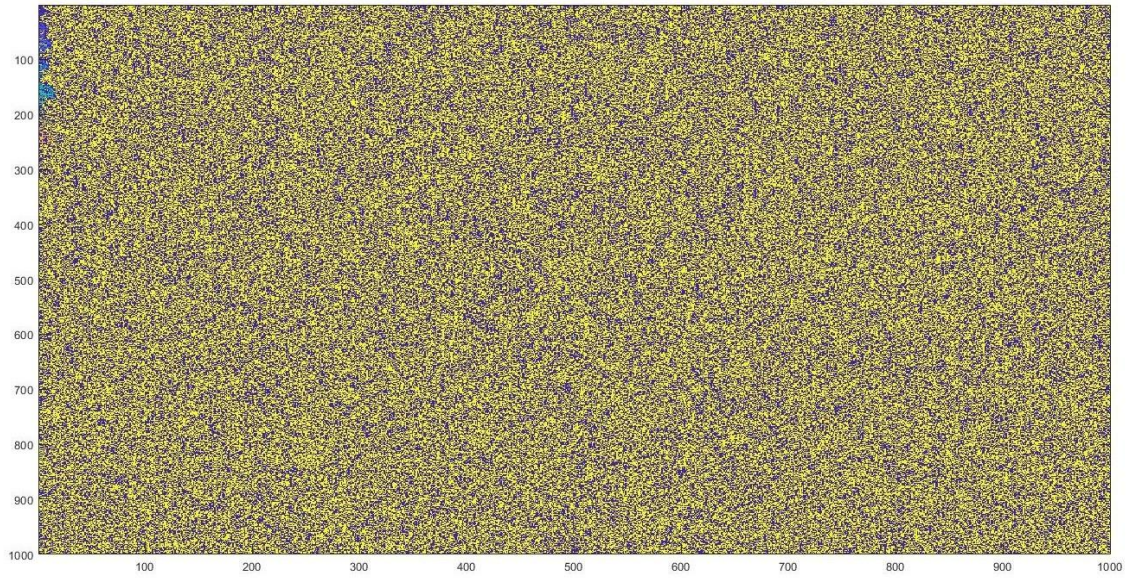


Şekil 4.7. En büyük Lyapunov üsteli grafiği

Elde edilen sonuçlara göre, veri sayısı yeterli sayıya ulaştığında (yaklaşık 7000) Lyapunov üsteli 0'dan büyük değer almaktadır. Tüm veri seti için elde edilen en büyük Lyapunov üsteli değeri ise 0.10955'tir. Buna göre ölçülen elektrik alan değerlerinin çok az sayıda örnek alınmadığı sürece kaotik davranış sergilediğini söylemek mümkündür. Bu analiz yardımıyla, elektromanyetik alan ölçüm değerlerinin RSÜ yapısı içerisinde kullanılabilirliği gösterilmiştir.

4.3.5. Hoshen-Kopelman Algoritmasıyla Rastgelelik Tespiti

Hoshen-Kopelman (HK) algoritması, birleşme bulma algoritmaları türünde bir analiz algoritmasıdır. Hoshen-Kopelman Algoritması, rastgele sayı üreteçleri için bir test olarak kullanılabilir. HK algoritması, veri kümelerini ızgara yapısı içerisinde hücrelerin organize bir ağ olduğu bölümleri tespit ederek işaretleme mantığına dayanan bir sistemdir [200]. Bu sistem, bitişik hücrelerde kümelenmeyi tanımlamak için etkili bir yöntemdir. Joseph Hoshen ve Raoul Kopelman tarafından yıllar önce önerilse de hala literatürde kendisine kullanım alanı bulmaktadır [201]. Rastgelelik kaynağı olarak kullanılan ölçüm değerlerinin HK algoritmasına göre kümelenme işaretlemeleri Şekil 4.8'de gösterilmiştir.



Şekil 4.8. HK algoritması kümelenme etiketleri

Yapılan analiz neticesinde kümelenme oranının oldukça fazla olduğu görünmektedir. Yani kaynak veriler büyük topluluklar halinde kümelenemeyerek küçük topluluklar oluşturabilmiştir. Bu durum kullanılan verilerin HK algoritmasına göre rastgelelik şartını sağladığını göstermektedir. Toplam 1 milyon örnekle yapılan test başarıyla sonuçlanmıştır.

4.4. Önerilen Şifreleme Sistemi Tasarımları

Tez çalışması kapsamında, farklı şifreleme sistemi tasarımları gerçekleştirilmiştir. Farklı şifreleme sistemi tasarlamanın belirli amaçları olmalıdır. Öncelikle özel bir veri grubu için özel tasarımlar yapılabilir. Burada veri çeşidine göre farklı şifreleme uygulamalarının etkinliği incelenebilir. Örneğin görüntü şifreleme sistemleri için çok farklı tasarımlar mevcuttur. Görsel olarak incelenebilen veri türlerinde şifreleme sistemindeki en ufak açıkta tüm veri elde edilemese bile şifreli olarak iletilmek istenen bilginin içeriğine erişmek mümkün olabilir. Bu yüzden ona uygun bir şifreleme yöntemi seçilmesi önemlidir. Ayrıca, veri iletiminin yapıldığı ortam da seçilecek şifreleme tekniğini belirleme de önemli bir parametre olacaktır. Kısmen kapalı bir haberleşme ortamında veri iletirken çok gelişmiş bir kriptografik sistem kullanmaya gerek yoktur. Veri erişimin açık olduğu ortamlarda ise gizlilik derecesi yüksek olan bir veri iletirken şifreleme sistemi iyi seçilmelidir. İletilen veri boyutu büyük ve sürekli iletim gerekiyorsa kriptografik sistemin hızlı çalışması da önemli bir parametre olacaktır. Anlık iletilmesi istenen veri hızından yavaş çalışan şifreleme sistemleri gecikmeye sebep olarak sistemde farklı sorunları doğurabilir. Kriptografik sistemlerdeki diğer bir parametre ise sistem gereksinimleridir. Oluşturulan haberleşme sisteminin sistem kaynaklarının durumuna göre de şifreleme sistemi belirlenebilir. Sistem

kaynaklarının kısıtlı olduğu mikro cihazlarda olabildiğince az sistem kaynağı tüketen şifreleme sistemi seçilmesi gerekmektedir.

Tüm bu şartlar göz önüne alındığında birçok farklı şifreleme sistemi olsa da güncel çalışmaların sürekli daha iyi bir sistem tasarımı gerçekleştirmeye çalıştığı görülebilir [67]. Kriptografik sistem hızını mikro düzeyde bile arttırabilen yeni bir sistem tasarlandığında tüm sisteme büyük etkileri olabileceği için büyük kullanım alanları bulabilmektedir. Yapılan tez çalışması kapsamında yukarıda belirtilen parametrelerin bir ya da birkaçını iyileştirmek için yeni şifreleme sistemi tasarımları yapılmıştır.

4.4.1. Şifreleme Algoritmalarında S-Kutusu Kullanımı

S-Kutuları genellikle blok şifreleme algoritmaları içerisinde kullanılan ve kullanıldığı algoritmaya doğrusal olmama özelliği veren yegane bileşendir. Bu nedenle iyi özelliklere sahip bir blok şifreleme algoritması geliştirebilmek için S-Kutularının kullanımı gereklidir. Yapılan birçok çalışmada farklı farklı S-Kutuları geliştirilmeye çalışılmıştır. S-Kutuları geliştirilirken de farklı yöntemler mevcuttur. Bu yöntemler içerisinde en popüler olan AES şifreleme standardında kullanılan ters dönüşüm yöntemidir. Tabi ki bu yöntem uzun süredir kullanıldığı için zaman içerisinde etkinliğini yitirebilir. Şifreleme sistemlerinin sürekli kendisini yenileme ihtiyacından gelen bu özellik yeni S-Kutularının tasarlanmasını da desteklemektedir. Bu amaçla son yıllarda özellikle kaos temelli çalışmalar öne çıkmıştır. Kaotik sistem çıkışından elde edilen S-Kutuları iyi karakteristik özellikler sergilemektedir [202].

S-Box tasarımlarında uyulması gereken 5 temel kriter bulunmaktadır. Bunlar;

1- Doğrusal olmama kriteri,
2- Bijektif olma özelliği,
3- Katı çığ kriteri,
4- Olası giriş/çıkış XOR dağılımı,
5- Çıkış bitlerinin bağımsızlığı kriteri,

olarak sıralanabilir. Bu kriterlere uygun S-Kutusu tasarlanması önem arz etmektedir. Ancak tüm bu kriterlere uygun olarak tasarlanan S-Kutuları kriptolojik uygulamalar için uygun olabilmektedir.

S-Kutularının ismi İngilizce “substitution” yani yer değiştirme kelimesinden gelmektedir. Yer değiştirme araçları olarak adlandırılan ve genellikle $n \otimes n$ boyutunda bir matris olarak tanımlanan S-Kutuları sistem içerindeki veriyi kendi yapısı içerisinde bulunan veriyle değiştirerek işlem sağlar. Şifreleme sistemlerindeki adımlar içerisinde yer değiştirme yapılan S-Kutusu, yine aynı kutunun tersi alınarak oluşturulan ters S-Kutusu matrisinin yardımıyla şifre çözme işleminde kullanıldığında herhangi bir hataya neden olmadan başarılı bir işlem sağlar. Şifreleme sistemleri

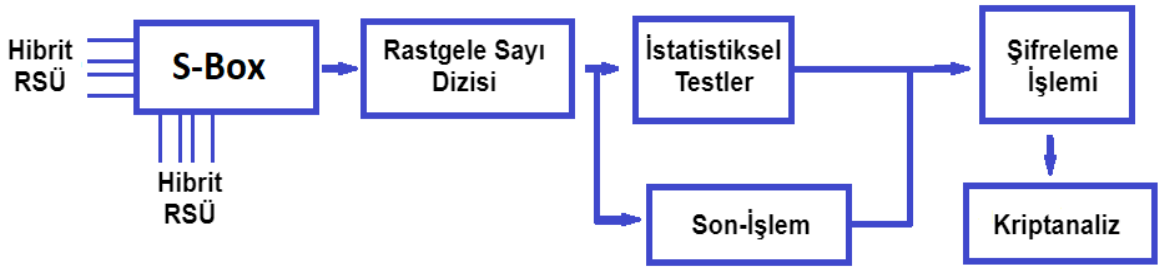
içinde doğrusal olmayan bir yapı olması sebebiyle S-Kutularının kullanımı oldukça yaygındır. Bu yapının sisteme kazandırdığı avantajlardan vazgeçmeyen araştırmacılar şifreleme sistemleri içinde yaygın şekilde kullanmaktadırlar [203].

Şifreleme algoritmalarından bağımsız olarak S-Kutuları rastgele sayı üreteci olarak da kullanılabilirler. Başlangıç değerleri elde bulunmak şartıyla 1'e 1 oranda yani ne kadar veri varsa aynı boyutta RSÜ çıkışı üreterek son-işlem algoritması olarak da kullanılabilirler. Yapılan tez çalışmasında S-Kutuları hem RSÜ sistem tasarımlarında hem de özgün şifreleme sistemi içerisinde kullanılmıştır.

4.4.2. Hibrit Rastgele Sayı Üreteci Tasarımı ve Sistem Detayları

GRSÜ ve SRSÜ tasarımlarından daha önce bahsedilmişti. Hibrit rastgele sayı üreteçleri her iki sistemi birlikte kullanan yapılardadır. Bilindiği gibi SRSÜ'ler belirli bir algoritma ile üretildikleri için başlangıç şartları ya da algoritma detayları öğrenildiğinde üretilen sayıların gizli kalması mümkün olmamaktadır. Bu amaçla özel algoritmalar yardımıyla başlangıç değerleri bir GRSÜ'den alınarak oluşturulan Hibrit tasarımlar literatürde etkinlik ve gizlilik açısından kriptolojik uygulamaların gereksinimlerini karşılamaktadır [204].

Örneğin, Trivium algoritması bir GRSÜ tasarımı üzerine uygulandığında hem sistemin istatistiksel testlerde daha başarılı olmasını sağlamış hem de kolay uygulanabilirliğiyle gerekli bitlerin hızlı bir şekilde üretilmesini sağlamıştır [25].



Şekil 4.9. Önerilen görüntü şifreleme sisteminin blok şeması

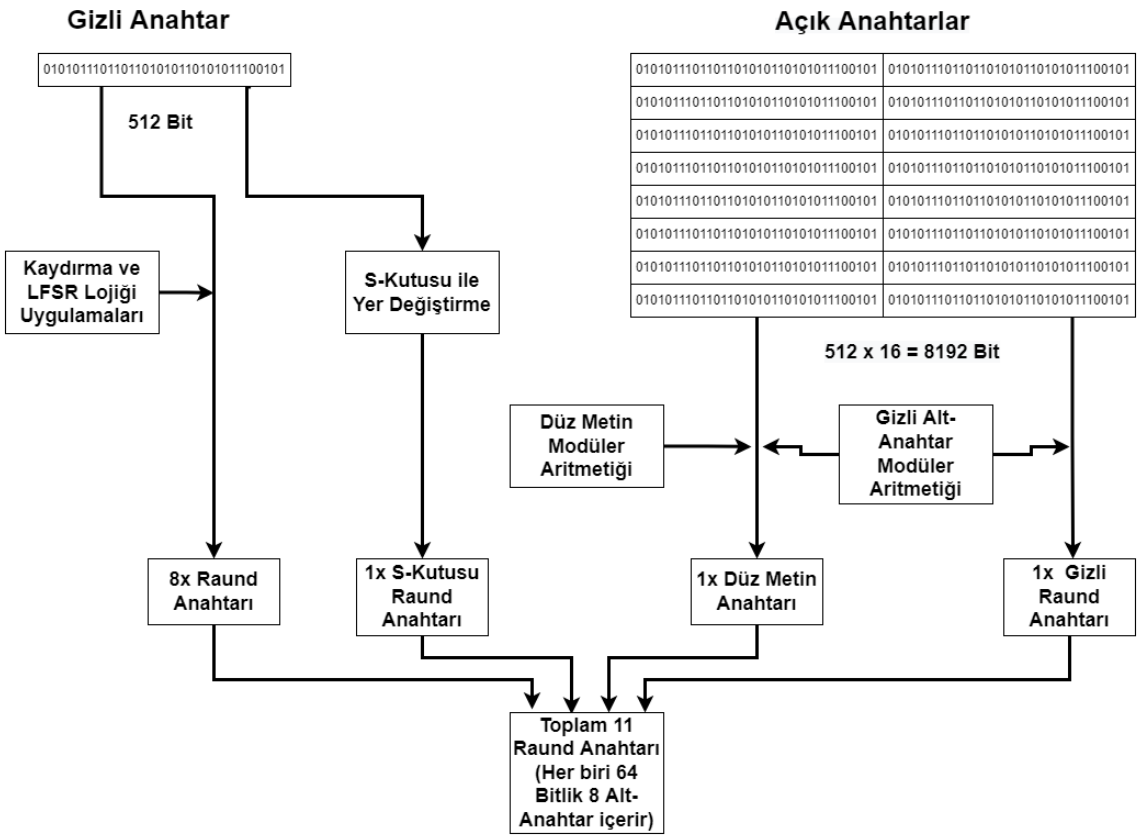
Şekil 4.9'da, tasarlanan sistemin blok şeması görülmektedir. Tasarladığımız bu sistem rastgele sayı üreteci çıktılarını Rijndael S-Kutusunda kullanarak şifreleme uygulamaları için uygun bir rastgele sayı dizisi elde edecektir. Çıktı olarak elde edilen rastgele sayı dizisi istatistiksel olarak değerlendirilerek ihtiyaç var ise son-işlem algoritmaları yardımıyla istatistiki değerleri iyileştirilecektir. Değerlendirmeyi başarılı bir şekilde geçen bit dizileri literatürde mevcut olan şifreleme algoritmalarıyla ya da kendi geliştirilen şifreleme algoritmalarıyla kullanılarak veri şifreleme işleminde kullanılacaktır.

S-Kutusu temelli olacak sistem veri seçiminde 2 farklı Hibrit rastgele sayı üretecinden gelen bilgileri kullanacaktır. Farklı gerçek rastgele sayı üreteçleriyle sözde rastgele sayı üreteci algoritmalarını kullanacak olan sistem gerekli olduğu taktirde son-işlem algoritmalarından da faydalanacaktır. İstatistiksel testler yardımıyla Hibrit RSÜ'ler için en uygun tasarımlar belirlenerek şifreleme sistemleri içerisinde kullanılacaktır. Geliştirilen şifreleme algoritması farklı kriptanaliz teknikleriyle test edilerek güvenli bir sistem oluşturulmaya çalışılacaktır.

4.4.3. Önerilen Yeni Şifreleme Sistemi

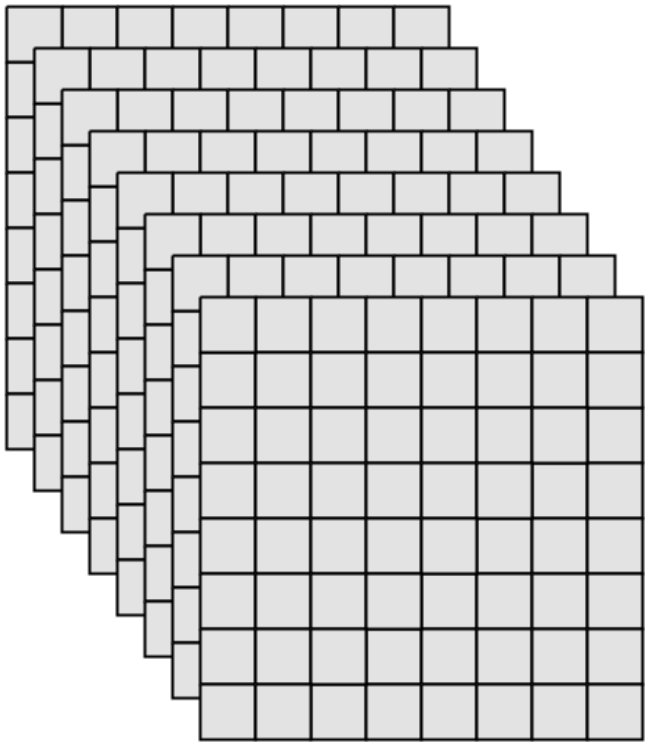
Yapılan çalışmalar sonucunda, her türlü veriyi şifreleyebilecek özgün bir simetrik blok şifreleme algoritmasının kendisine kullanım alanı bulabileceği belirlenmiştir. Tasarlanan sistemin kriptanaliz yöntemleri içerisinde belirtilen tüm analizlerde başarılı olmasının yanı sıra yüksek bir sistem karmaşasına sahip olmaması da önemli bir parametre olacaktır. Bu kapsamda güncel 256-bit'e kadar olan anahtarlı blok şifreleme algoritmalarının yıllar içerisinde artan bilgisayar işlem gücü karşısında savunmasız kalabileceği düşünülerek 512-bit anahtarlı bir sistem tasarlanmıştır. Önerilen sistem iki aşamadan oluşmaktadır. Şifreleme anahtarı üretme ve şifreleme ana işlemi olarak belirlenen sistem aşamaları birbirinden bağımsız çalışabilmesi sebebiyle işlem hızını önemli ölçüde arttıracaktır.

Şifreleme anahtarı üretme sisteminin ana şeması Şekil 4.10'da verilmiştir. Buna göre, üretilecek olan 11 farklı şifreleme raund anahtarının üretilme yöntemleri belirlenmiştir.



Şekil 4.10. Önerilen şifreleme algoritmasının raund anahtarı üretme tekniği

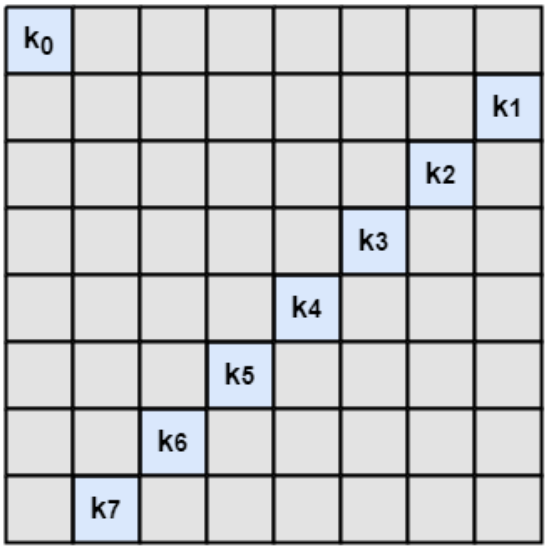
Burada gösterilen 11 raund anahtarının her biri 64 bitlik 8 ayrı alt-anahtar içermektedir. Toplamda 88 adet alt anahtar yeni önerilen şifreleme sistemi içerisinde yer alacaktır. 8 alt-anahtar içeren her bir raund anahtar 8x8x8'lik şifreleme anahtarı küpleri şeklinde tasarlanmıştır. Her bir raund anahtarına ilişkin bitlerin küp yapısı içerisindeki dağılımları Şekil 4.11'de gösterilmiştir.



Şekil 4.11. 8x8x8 raund anahtarı küp yapısı

Gösterilen şekilde her bir kare bir bit'i temsil etmek üzere gösterilen her bir katman 8x8=64 bit'ten oluşmaktadır. Toplamda 8 katman içeren sistem 512 bitlik raund anahtarını oluşturacaktır. Burada gösterilen anahtar katmanları kendi içlerinde ve birbirleri arasında farklı işlemler yapılmak üzere bu şekilde belirlenmiştir.

İlk sekiz raund anahtarı oluşturulurken orijinal anahtar üzerinde kaydırmalar ve lojik işlemler yapılarak diğer anahtarlar elde edilmektedir. Burada temel hedef çok fazla işlem karmaşası yaratmadan farklı raund anahtarları elde etmektir. Yapılan işlemler neticesinde birbirinden tamamen farklı anahtarlar elde edilemeyebilir ancak bu durum şifreleme sisteminin güvenliğini etkilemiyorsa bir problem teşkil etmeyecektir. Raund anahtarları elde edilirken ilk turda 15 bit'lik daha sonraki her turda da 7 bit'lik her bir katman kendi içerisinde kaydırılır. Bu durum Şekil 4.12'de gösterilmiştir.

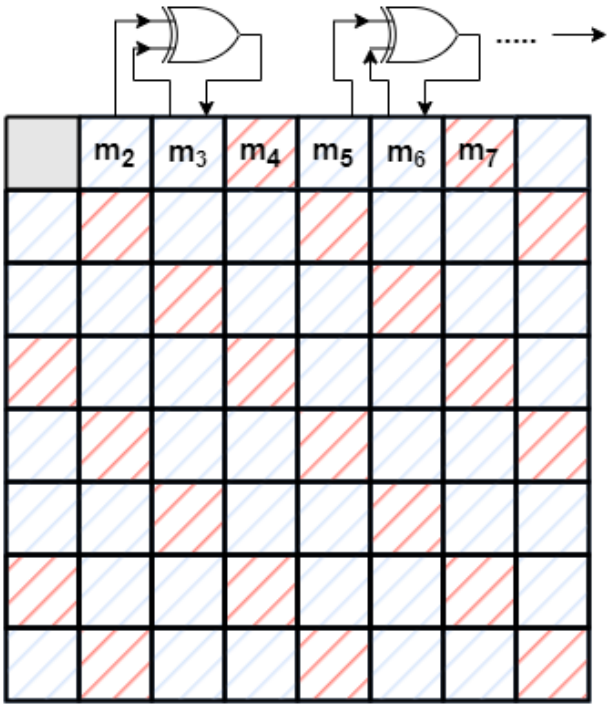


Şekil 4.12. Kaydırma işlemleri sonrası ilk bit pozisyonları

Burada kaydırmalar yapılırken göz önünde bulundurulan nokta, şifreleme sistemi içerisinde döngüler çalışırken işlem yapılan bit konumlarının sürekli sabit olması zaman zaman bir önceki döngüdeki işlemi geri alacak şekilde döngü sağlayabilir. Bunu engellemek için birçok şifreleme sisteminde kaydırma işlemleri kullanılır ancak bit'lerin kaydırılması tek başına yeterli olmayacaktır. Bu yüzden her bir katmandaki bitlere denklem 4.16'da gösterildiği şekilde XOR lojik operasyonu uygulanır.

$$m_{n+1} = (m_n \oplus m_{n+1}) \tag{4.16}$$

Bu işlemler yapılırken, Doğrusal Geri-Beslemeli Kaydırmalı Kaydediciler (Linear Feedback Shift Register) mantığından yola çıkılarak katmanların içerisinde bir rastgelelik sağlamak amaçlanmıştır. Zaten LFSR uygulamaları rastgele sayı üreteci tasarımlarında kullanıldığında başarılı sonuçlar elde edildiği için bu yapılardan faydalanılmıştır. Şekil 4.13'te uygulanan XOR işlemleri grafiksel olarak açıklanmıştır.



Şekil 4.13. Katmanlar içerisinde XOR işleminin uygulanışı

Burada toplam her bir katman için 21 işlem uygulanmış olacaktır. Mavi renkli kutucuklar işleme giren kırmızı renkli kutucuklar ise işleme girmeyecek bitleri temsil etmektedir. Bu yöntemle raund anahtarları katmanları arasında gerekli rastgeleliğin sağlanıldığından emin olunmasının yanı sıra çok fazla işlem karmaşasına girilmediği için sistem hızı da minimum ölçüde etkilenecektir. Eğer anahtarların fazla birbirine benzediği düşünülürse ve bu bir güvenlik açığı yaratırsa bu aşamada farklı işlemler uygulanarak bu sorun kolaylıkla giderilebilecektir.

Bu işlemler sonucunda ilk 8 raunda anahtarı hazırlanmış olur. Kalan 3 raund anahtarı üretilirken izlenen yöntemler burada açıklanmıştır.

Gizli şifreleme anahtarı alınarak S-Kutusu üzerinde yer değiştirme işlemine tabi tutularak yeni bir bit dizisi elde edilebilir. Burada kullanılan Rijndael S-kutusu Şekil 4.14'de gösterilmiştir.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Şekil 4.14. Rijndael S-kutusu değerleri

Rijndael S-kutusu 16x16 yapıdadır ve toplamda 256 değer içermektedir. Bu S-Kutusu üzerinde işlem yaparken 16 satır ve 16 sütun içerisinde seçim yapabilmek için gizli anahtarın her 4 bit'lik kısmı bir sütun ya da bir satır seçmek için kullanılır. Belirlenmen satır ve sütunda bulunan değer 8 bitlik bir veri ortaya koyduğundan gizli anahtardan toplam 4+4=8 bit için S-Kutusundan 8 bit değer alınmış olur. Herhangi bir veri kaybı olmadan 512 bit'lik gizli anahtar, 512 bit'lik raund anahtarına çevrilmiş olur. Böylece 9. Raund anahtarı da üretilmiş olur.

Geride kalan son iki raund anahtarını elde etmek için modüler aritmetik işlemlerinden ve önceden belirlenmiş olan 16 farklı açık anahtardan faydalanılır. Açık anahtarların her biri 512 bit'tir. Tüm herkese açık olan bu anahtarların herkesin ulaşabileceği bilinse de zaman zaman değişiklik yapılması şifreleme sistemi güvenliği için önerilmektedir.

Son iki raund anahtarı üretilirken şifrelenecek olan verinin ve gizli anahtarların modüler aritmetik işlem sonuçlarından faydalanılır. Daha önce hazırlanmış olan raund anahtarlarının mod16 değerleri ve şifrelenecek olan metinin mad16 değerlerine göre 16 adet açık anahtar içerisinden seçimler yapılır. Bu seçimler sonucunda, belirlenen en az 5 farklı açık anahtar birbiriyle XOR işlemine tabi tutularak son raund anahtarları üretilebilir. Denklem 4.17'de bu işlemin detayları gösterilmiştir.

$$Raund - Anahtarı_{10} = (PK_a \oplus PK_b \oplus PK_c \oplus PK_d \oplus PK_e) \quad (4.17)$$

Burada PK değerleri açık anahtarları alt indisler ise hangi anahtarın seçildiğini göstermektedir. Bu işlem sonucunda elde edilen raund anahtarı açık anahtarlardan tamamen farklı bir yapıda olacaktır. Ayrıca şifreleme döngüsü içerisinde kullanılacağı için, hangi anahtarların kullanıldığını belirlemek neredeyse imkânsız olacaktır.

Bu aşamaların gerçekleştirilmesinden sonra, elde edilen 11 raund anahtarı şifreleme sistemi için hazır olacaktır. Bundan sonraki aşamada şifreleme işleminin kendisi gerçekleşecektir. Önerilen algoritma içerisinde anahtar üretim sisteminin ve şifreleme ana işleminin ayrı yapılarda bulunarak birbirinden bağımsız olması önemli avantajları beraberinde getirecektir. Şifreleme ana gövde işlemi 11 anahtarı 11 raundda kullanmaktadır. Şifreleme adımları aşağıda gösterilmiştir.

Raund 1: Şifreleme işleminin başlangıç adımında ilk raund anahtarı alınarak şifrelenecek metin ile XOR işlemine tabi tutulur. Bu raundda başka bir işlem yapılmaz.

Raund 2: İkinci adımda ise ilk adımda üretilen şifrelenmiş metin öncelikle katmanlar arası geçişte yapabilecek şekilde 47 bit kaydırılır. Böylece 8x8'lik her bir katman değişmiş olur. Ayrıca 2. Raund itibariyle her raundda katmanlar kendi arasında karıştırılarak karmaşıklık arttırılmış olur. Daha sonra bu raundun anahtarıyla XOR işlemi yapılarak bu aşamada bitirilir. Tablo 4.2'de her raundda değiştirilen katmanlar gösterilmiştir.

Tablo 4.2. Şifreleme adımları içerisinde yapılan katman değişiklikleri

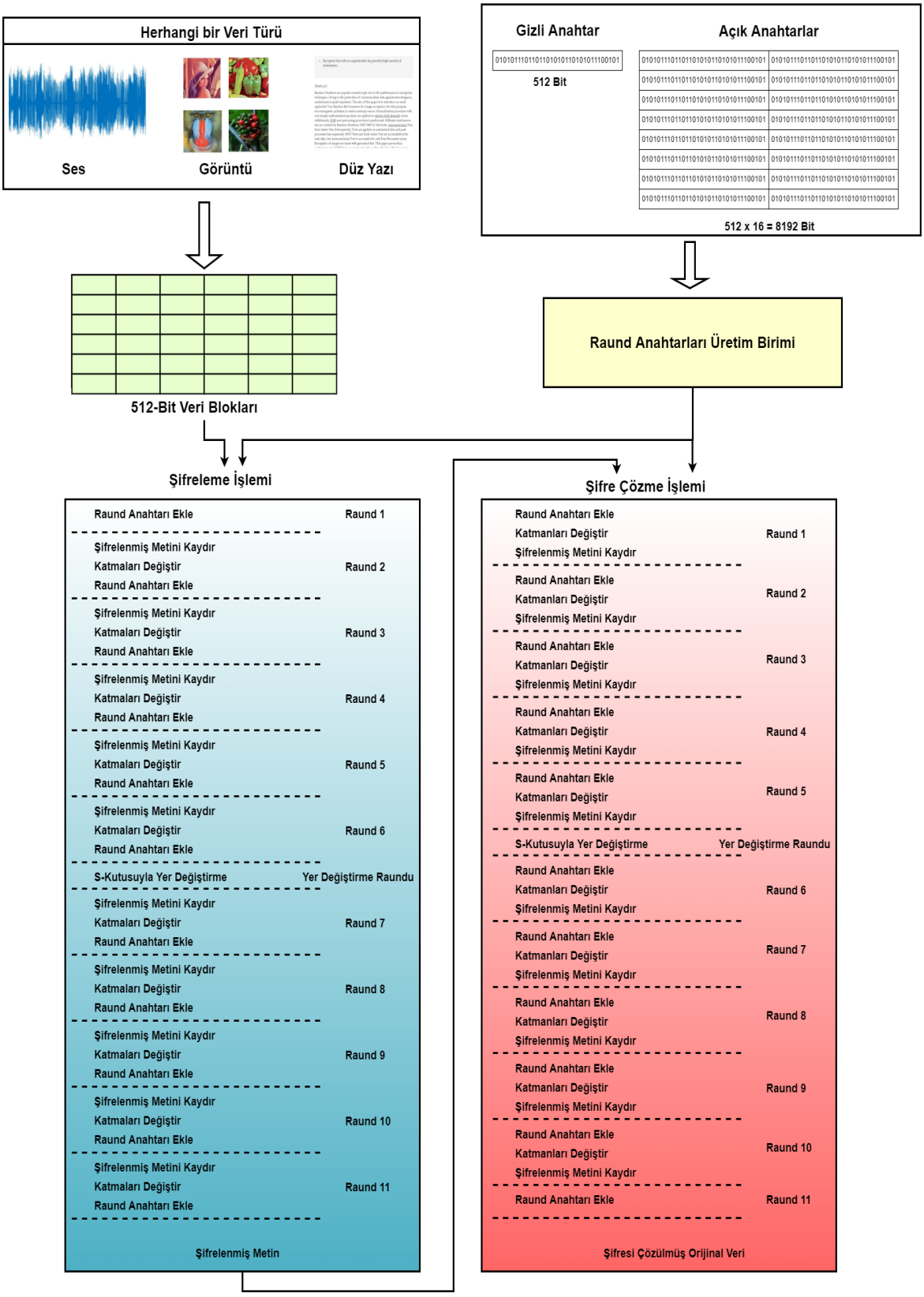
Raund Numarası	1. Katman	2. Katman	3. Katman	4. Katman	5. Katman	6. Katman	7. Katman	8. Katman
Raund-1	1	2	3	4	5	6	7	8
Raund-2	2	4	6	8	1	3	5	7
Raund-3	4	8	3	7	2	6	1	5
Raund-4	8	7	6	5	4	3	2	1
Raund-5	7	5	3	1	8	6	4	2
Raund-6	5	1	6	2	7	3	8	4
Raund-7	6	3	5	7	8	1	4	2
Raund-8	3	7	1	8	6	4	2	5
Raund-9	2	4	6	8	1	3	5	7
Raund-10	4	8	3	7	2	6	1	5
Raund-11	8	7	6	5	4	3	2	1

Raund 3: Raund 3'ten itibaren aynı şekilde 47 bit'lik katmanlar arası bit kaydırma işlemi daha sonra katmanların yer değiştirmesi yukarıdaki tabloya göre gerçekleştirilir. Örnek vermek gerekirse, ilk raundda zaten tüm katmanlar aynı numaralarıyla işleme alınmıştır. Yani tüm katmanlar aynı şekilde bekler. 2. Raundda 1. Katman artık 2. Katman olmuştur. 2. Katmanda 4. Katman olarak yerini alır bu şekilde tüm katmanların yeri değiştirilmiş olur. Her raundda kullanılan şifreleme anahtarlarının listesi Tablo 4.3'te verilmiştir.

Tablo 4.3. Şifreleme adımları içerisinde kullanılan şifreleme anahtarları

Raund Numarası	Şifreleme Anahtarları
Raund-1	Orijinal Gizli Anahtar
Raund-2	S-Kutusu ile Üretilen Anahtar
Raund-3	Açık Anahtarlar Yardımıyla Üretilen Anahtar-1
Raund-4	Açık Anahtarlar Yardımıyla Üretilen Anahtar-2
Raund-5	LFSR Mantığıyla Gizli Anahtardan Üretilen Anahtar-1
Raund-6	LFSR Mantığıyla Gizli Anahtardan Üretilen Anahtar-2
Raund-7	LFSR Mantığıyla Gizli Anahtardan Üretilen Anahtar-3
Raund-8	LFSR Mantığıyla Gizli Anahtardan Üretilen Anahtar-4
Raund-9	LFSR Mantığıyla Gizli Anahtardan Üretilen Anahtar-5
Raund-10	LFSR Mantığıyla Gizli Anahtardan Üretilen Anahtar-6
Raund-11	LFSR Mantığıyla Gizli Anahtardan Üretilen Anahtar-7

Raund 3'den sonra tüm aşamalarda aynı işlemler yapılmaktadır. Yalnızca 6. raunddan sonra yapılan yer değiştirme işleminde S-kutusuyla 6. Raund çıkışında elde edilen şifrelenmiş metine yer değiştirme işlemi uygulanır. Böylece S-Kutularının sağladığı doğrusal olmama özelliğinden de faydalanılır. Diğer raundlar aynı şekilde işlemlere devam eder. Tüm şifreleme adımları Şekil 4.15'te gösterilmiştir.



Şekil 4.15. Önerilen şifreleme algoritmasının genel yapısı

Şifreleme işlemleri ardından, iletilen veri şifreleme işlemlerinin tam tersi yönde yapılan uygulamalarla şifre çözme işlemiyle geri döndürülmüş olur. Burada işlemler ter yönde yapıldığı için katman değişimleri de tersine döndürülmüş olur. Tablo 4.4'de şifre çözme uygulaması için yapılması gereken katman değişimleri belirtilmiştir.

Tablo 4.4. Şifre çözme adımları içerisinde yapılan katman değişiklikleri

Raund Numarası	1. Katman	2. Katman	3. Katman	4. Katman	5. Katman	6. Katman	7. Katman	8. Katman
Raund-1	8	7	6	5	4	3	2	1
Raund-2	7	5	3	1	8	6	4	2
Raund-3	5	1	6	2	7	3	8	4
Raund-4	3	7	1	6	8	5	2	4
Raund-5	6	8	2	7	3	1	4	5
Raund-6	2	4	6	8	1	3	5	7
Raund-7	4	8	3	7	2	6	1	5
Raund-8	8	7	6	5	4	3	2	1
Raund-9	7	5	3	1	8	6	4	2
Raund-10	5	1	6	2	7	3	8	4
Raund-11	1	2	3	4	5	6	7	8

Katman değişimlerinin yanı sıra, bit kaydırma işlemleri de tam tersi yönde yapıldığında sorunsuz bir şekilde şifre çözme işlemi gerçekleşerek orijinal veriyi elde etmek mümkün olur. Tablo 4.5'te önerilen şifreleme algoritmasının literatürdeki en popüler blok şifreleme algoritmaları olan DES ve AES'e göre karşılaştırmalı özellik tablosu verilmiştir.

Tablo 4.5. Önerilen şifreleme algoritmasının karşılaştırma tablosu

Algoritma Karakteristiği	DES	AES	Önerilen
Blok Boyutu	64-Bit	128/192/256-Bit	512-Bit
Raund Sayısı	16	10-14	11
Anahtar Boyutu	64-Bit	128/192/256-Bit	512-Bit
Alt-Anahtar Sayısı	16	44	88
Alt-Anahtar Boyutu	48-Bit	32-Bit	64-Bit
Her Raundda Kullanılan Alt-Anahtar	1 Anahtar	4 Anahtar	8 Anahtar
Şifrelenmiş Metin Boyutu	64-Bit	128/192/256-Bit	512-Bit

Karşılaştırma tablosuna göre, en son ve geçerli blok şifreleme algoritması olan AES algoritmasına göre anahtar boyutu 512 bit'e çıktığı için çok daha güvenlikli bir kriptografik sistem

geliştirildiği söylenebilir. Ayrıca bir önceki standart olan DES'den AES algoritmasına geçiş sağlandığında geliştirilen karakteristik özellikler önerilen algoritmayla da örtüşmektedir. Bu açıdan, önerilen algoritmanın blok şifreleme algoritması olarak kullanılmasının uygun olabileceği görülmektedir.

4.4.4. Önerilen Piksel Karıştırma Yöntemi

Görüntü şifreleme uygulamalarında, ön-işlem olarak piksel karıştırma algoritmalaranının uygulanması algoritmaların güvenilirliğini arttırmaktadır. Şifreleme uygulamalarının yerini tutmasa da piksel karıştırma işlemi uygulanan görüntü şifreleme sistemlerinin daha güvenilir olduğu düşünülmektedir [205–208]. Bu amaçla rastgele sayı üreteci tabanlı bir rastgele permütasyon oluşturularak piksel karıştırma işlemi uygulanmıştır. 512x512 piksel boyutundaki bir görüntü için Şekil 4.16'da belirtilen şekilde piksel numaraları belirlenmiştir.

	1	2	3	....	....	512
1	1	513	1025	.....	.....	261633
2	2	514	1026	.....	.....	261634
3	3	515	1027	.....	.....	261635
4	4	516	1028	.....	.....	261636
5	5	517	1029	.....	.....	261637
6	6	518	1030	.....	.....	261638
....	.....	.....	.....	.....	.....	.....
....	.....	.....	.....	.....	.....	.....
....	.....	.....	.....	.....	.....	.....
....	.....	.....	.....	.....	.....	.....
....	.....	.....	.....	.....	.....	.....
....	.....	.....	.....	.....	.....	.....
512	.....	.....	.....	.....	.....	.....

Şekil 4.16. 512x512 piksel görüntü için piksel numaralandırması

Bu şekilde belirtilen bir görüntü için 262144 adet pixel numarası gösterilmiştir. Bit düzeyindeki rastgele sayı dizileri 18 bitlik sayılara çevrildiklerinde tüm piksel numaralarını kapsayacak şekilde sayılar elde edilmiş olur. Bu sayılarla Şekil 4.17'deki gibi bir matris oluşturulabilir.

	1	2	3	4	5	6	7	8	9
1	31333	202143	177364	26568	229299	6448	235999	259998	98654
2	68237	55423	217470	257435	119811	119291	83747	1777	206958
3	34816	37592	75649	1414	221240	218973	117056	114412	150475
4	198865	26598	144464	5180	45762	75441	155508	148554	176507
5	164070	223025	48490	167478	5985	56573	55248	171216	244700
6	31416	34329	104132	16851	242087	75179	7946	56679	124389
7	140989	79731	177944	4756	68400	223290	121763	206331	71025
8	46776	5268	204665	163985	160323	114419	46498	117607	42716
9	65320	182801	253199	31145	158595	55737	2473	15466	95994
10	236177	203560	93110	174970	93012	41746	29820	202857	169393
11	124366	216681	151530	172066	17839	58934	153656	498	69606
12	30731	57717	166184	188735	27845	138550	171268	135294	118614
13	35281	123635	61386	96175	65775	174831	84736	1651	210782

Şekil 4.17. 512x512 piksel görüntü için piksel numaralandırması

Böylece rastgele permütasyon matrisini elde etmek mümkün olur. Burada en önemli nokta, numaralar yerleştirilirken aynı numaranın birden fazla yerleştirilmesinin önüne geçilmesidir. Böylece her pikselin karıştırma işlemiyle gideceği tek bir yer olacaktır. Belirtilen piksel numaralarına göre orijinal görüntünün piksellerinin yerleri değiştirildiğinde piksel karıştırma işlemi tamamlanacaktır.

4.5. Kriptanaliz Yöntemleri

Kriptanaliz terimi, şifrelenmiş olan metinlerin çözümü için gizli anahtarları tespit etme, şifreleme sistem yapısını öğrenme ya da şifrelenmiş metinin bir kısmının veya tamamının elde edilmesi gibi amaçlarla ortaya çıkmıştır. Klasik kriptanaliz yöntemlerinin temel amacı, şifreyi kırmak ya da gizlenen bilgiye ulaşmaktır. Anahtarı ortaya çıkarmayı amaçlayan yöntemler sadece anlık verinin elde edilmesi dışında gelecekte gönderilecek bilgilere de erişim sağlayabilir. Modern kriptanaliz teknikleriyse şifreleme sisteminin her türlü zaafını genellikle bilgisayar destekli analizlerle elde etmektedir. Tablo 4.6'da klasik yöntemler ve bu yöntemler yardımıyla elde edilebilecek bilgiler verilmiştir.

Tablo 4.6. Kriptanaliz yöntemleriyle elde edilmeye çalışılan bilgiler

Kriptanaliz Yöntemi	Elde Edilmeye Çalışına Bilgiler
Şifreli Metin Saldırısı	Şifreleme algoritması yapısı Şifrelenmiş metinin çözülmesi
Bilinen Metin Saldırısı	Şifreleme algoritması yapısı Şifrelenmiş metinin çözülmesi Aynı anahtarla oluşturulan tüm metinler
Seçilen Şifreli Metin Saldırısı	Şifreleme algoritması yapısı Şifrelenmiş metinin çözülmesi Seçilen anahtarla üretilen tüm metinler
Seçilen Metin Saldırısı	Şifreleme algoritması yapısı Şifrelenmiş metinin çözülmesi Seçilen anahtarla üretilen tüm metinler

Burada gösterilen saldırı yöntemleri, kriptanalizin temel hedeflerini gösterse de günümüzde kullanılan şifreleme yöntemleri bu parametreler ışığında farklı farklı testlerle değerlendirilmektedir. Modern sistemler yine aynı temel üzerinde ilerlese de her bir şifreleme tekniği için farklı yöntemler ele alınmalıdır. İlerleyen başlıklarda bazı kriptanaliz yöntemlerinin işleyiş mantığı ve güvenilir bir şifreleme için gerekliliklerden bahsedilecektir.

4.5.1. Anahtar Boyutu ve Uygulama Etkinliği

Anahtar boyutu, özellikle anahtarı tahmin etmeye yönelik kaba kuvvet ataklarına karşı sistemin direnci belirleyen, şifreleme sistemi için en önemli parametrelerden biridir. Çünkü gizli anahtarlı şifreleme sistemlerinde algoritma yapısı herkes tarafından bilinebileceği için gizli anahtarın bilinmesi tüm sistemin kriptolojik saldırılara karşı savunmasız olduğunu gösterecektir. Ancak iyi seçilmiş bir anahtarın belirlenmesi o kadar da kolay değildir. Rastgelelik şartlarını sağlayan bir şifreleme anahtarını doğrudan tahmin edebilmek oldukça zordur. Sadece kaba kuvvet saldırılarıyla tüm ihtimaller denenerek şifreleme anahtarı bulunabilir. Bu noktada şifreleme anahtarı boyutunun önemi ortaya çıkmaktadır. Teorik olarak tüm ihtimallerin denenmesi çok uzun zaman alırsa bu işlem için kaba kuvvet saldırıları etkili olmayacaktır. Tablo 4.7'de anahtar bit sayısının basit bir işlemci tarafından yaklaşık olarak tespit edilme süreleri gösterilmiştir. Burada işlemci teknolojileri ilerledikçe her geçen yıl işlem gücü daha da artsa da yaklaşık olarak seçilecek anahtar boyutuna göre tahmin sürelerindeki değişimler gösterilmiştir.

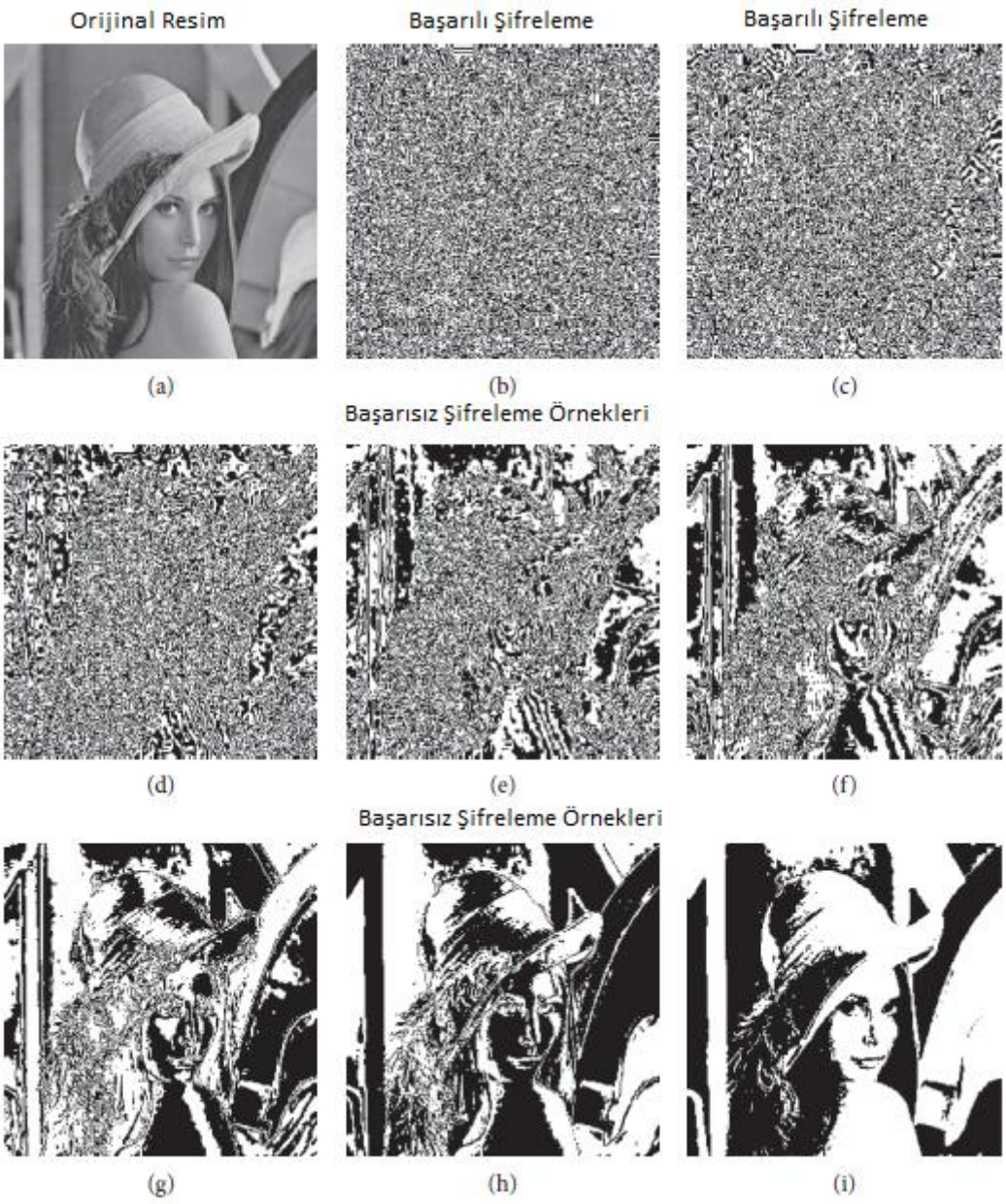
Tablo 4.7. Anahtar boyutuna göre yaklaşık anahtar tahmin süreleri

Anahtar Uzunluğu	Anahtar Boyutu	Şifre Tahmini için Yaklaşık Gereken Süre
32 Bit	$2^{32} = 4.3 \times 10^{9}$	2,15 milisaniye
64 Bit	$2^{64} = 7.2 \times 10^{16}$	10 saat
128 Bit	$2^{128} = 3,7 \times 10^{38}$	$5,4 \times 10^{8}$ yıl
256 Bit	$2^{256} = 1,15 \times 10^{77}$	$1,7 \times 10^{57}$ yıl

Görüldüğü gibi, söz konusu sistem için 32 bitlik anahtarı tahmin etmek milisaniyeler sürerken 128 bit ve üstü bir anahtara sahip sistem yıllar içerisinde kırılabilmektedir. Tabi ki güncel gelişmiş bilgisayarlar yardımıyla bu süreler çok daha az olabilse de anahtar boyutu ne kadar fazla olursa sistem bu saldırılar için o kadar dirençli olacaktır [29]. Burada önemli olan konu, anahtar boyutundaki her bir bit'lik artışın tahmin için geçen süreye katlanarak yansımasıdır. Ancak anahtar boyutunu gereksiz olarak büyük belirlemek de sistem karmaşasını arttırarak algoritmanın etkin çalışmasını etkileyebilir. Bunun sonucunda, şifreleme ve şifre çözme süreleri gereksiz yere uzayabilir. Bu yüzden optimal anahtar boyutunun tespit edilerek gerçekleştirilmesi sistemin sağlıklı çalışmasını sağlar.

4.5.2. Histogram Analizi

Şifrelenmiş veriler için sıklıkla uygulanan bir analizdir. Özellikle görüntü şifreleme alanında, görüntünün karmaşıklaşması önemlidir. Yani verinin temeli olan bit düzeyini inildiğinde şifrelenmiş bir veri topluluğunun olabildiğince eşit sayıda "1" ve "0" sayısına sahip olması istenir. Şifrelenmiş tüm verilerde eşit bir dağılım olursa bu verinin neye ait olduğunu belirlemek zorlaşacaktır. Şekil 4.18'de başarılı ve başarısı düşük resim şifreleme uygulamaları literatürde sıklıkla kullanılan örnek Lena resmi üzerinde gösterilmiştir.



Şekil 4.18. Örnek resim üzerinde farklı şifreleme uygulamaları sonucunda oluşabilecek görüntüler

Örnekte de görüldüğü gibi, karmaşık bir görüntüye sahip olmayan görüntüler için gizlenmek istenen resmi tahmin etmek zor olmamaktadır. Görüntü şifreleme uygulamalarında bu yüzden histogram analizi önem arz etmektedir [66]. Şifreleme işlemleri sonucunda ortaya çıkan görüntünün düzgün bir histogram dağılımına sahip olmaması durumunda, tam olarak şifrelenen veri belirlenemese bile bu şekilde gerçekleştirilen şifreleme sisteminin güvenlik konusunda zaafları olduğu varsayılır.

4.5.3. Korelasyon Katsayısı Analizi

Korelasyon katsayısı, bağımsız değişkenler arasındaki farklı türdeki ilişkileri belirlemek için kullanılan bir matematiksel analiz yöntemidir. Bu katsayı "1" ile "-1" arasındaki değerlerden oluşmaktadır. Pozitif değerlerle elde edilen ilişki doğrusal yöndeki bir ilişkiyi, negatif değerlerle elde edilen ilişki ise ters taraflı bir doğrusal ilişkiyi tanımlamaktadır.

Şifreleme uygulamalarında korelasyon katsayısı analizi kullanılırken, veri dosyaları arasında herhangi bir ilişki olup olmadığını belirlemeye çalışılır. İki farklı şekilde bu analizin kullanımı bulunmaktadır. Öncelikle şifrelenecek olan orijinal veriyle şifrelenmiş veri arasındaki korelasyon analizi yapılabilir. Ancak bu durum en basit şifreleme sisteminde bile şifrelenmiş verinin orijinal veriden çok farklı bir yapıya dönüştürülmesi sebebiyle mantıklı olmamaktadır. Her türlü şifreleme sistemi orijinal veriyi şifreleyerek farklı bir yapı grubu oluşturabileceği için korelasyon katsayısı "0" değerinin yakınında olacaktır. Yani bu iki veri grubu arasında herhangi bir ilişki olmadığı söylenebilecektir.

Korelasyon katsayısı analizinin asıl kullanım alanı, görüntü şifreleme sistemlerinde komşu pikseller arasındaki ilişkinin incelenmesiyle ortaya çıkmıştır. Tabi ki bu analiz diğer veri gruplarının şifrelenmesinde de komşu veri gruplarının arasındaki ilişki incelenerek uygulanabilir. Görüntü işleme uygulamalarında herkes tarafından bilinen bir gerçek şudur; özellikle yüksek piksel çözünürlüğüne sahip görüntülerin komşu pikselleri genellikle birbirine benzemektedir. Bu durum oldukça normaldir. Örneğin, standart bir portre resmi ele alırsak göz bölgesi genelde aynı renk tonlarından, saçlar yine yukarıda aynı bölgede belirli renk tonlarından oluşmaktadır. Bu bölgelerde bulunan görüntü pikselleri de birbirine büyük ölçüde benzeyecektir. Normal bir görüntünün pikselleri arasında bu bağlamda bir benzerlik olması normal iken, şifrelenmiş görüntüler için bu istenmeyen bir durumdur. Şifrelenmiş bir görüntünün komşu pikselleri arasında orijinal görüntüye benzer şekilde bir ilişki bulunması bir güvenlik açığına işaret edebilir.

Bir görüntü şifreleme sisteminde, korelasyon katsayısı analizi yapabilmek için şifrelenmiş görüntünün komşu pikselleri arasındaki korelasyon katsayısı hesaplanmalıdır. Bu hesaplama 4.18, 4.19 ve 4.20'deki denklemler yardımıyla yapılabilir [209].

$$r_{x,y} = \frac{E((x - E(x)(y - E(y)))}{\sqrt{D(x)D(y)}} \tag{4.18}$$

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{4.19}$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N} (x_i - E(x))^2 \tag{4.20}$$

Böylece şifrelenmiş görüntü için üç farklı yönde, yatay, dikey ve çapraz olarak korelasyon katsayılarının hesaplanması bu analizi yapmak için yeterli olacaktır. Elde edilen katsayı değerleri "0"a yakın olduğunda ya da orijinal görüntüye göre farklılık içerdiğinde bu analizin başarılı olduğu söylenebilir.

4.5.4. Bilgi Entropisi Analizi

Bilgi entropisi değeri, sistem kompleksliğini gösteren bir parametre olarak öne çıkmaktadır. Shannon tarafından ilk kez 1948'de ortaya atılan bilgi entropisi kavramı, Shannon entropisi olarak da bilinmektedir. Genel bilgi entropisi denklemi 4.21'de verilmiştir.

$$H_m = \sum_{i=0}^{n} p(m_i) \log \frac{1}{p(m_i)} \tag{4.21}$$

Yıllar içerisinde farklı entropi tanımlamaları yapılmaya çalışılsa da Shannon entropisi özellikle görüntü şifreleme uygulamaları için yeterli olması sebebiyle en yaygın kullanım alanı olan analiz olmuştur [210].

Bilgi entropisi değeri, esasında iletişim sistemleri içerisinde üretilen bilginin oranını tespit etmek için ortaya koyulmuştur. Şifreleme uygulamaları içerisinde kullanımı ise şu temel mantığa dayanmaktadır. Eğer entropi değeri, toplam bilgi değeri büyüklüğüne yakınsa elde edilecek olan değerler birbirine çok yakın olacaktır. Bu da veri kümesinin rastgele olduğu sonucunu doğurabilir [211].

Görüntü şifreleme sistemleri için çalışmada kullanılan görüntüler, 8 bit'lik örnek görüntü işleme resimleri olduğu için hesaplanacak olan H_m değeri olabildiğince "8"e yakın olmalıdır. Bu şekilde bir hesaplama yapıldığında şifreleme sisteminin gerekli güvenlik koşullarını sağladığı söylenebilecektir.

4.5.5. Diferansiyel Saldırı Analizi

Diferansiyel saldırı kriptanalizi, yinelemeli saldırılar için en etkili yöntemlerden biridir. Anahtar yardımıyla şifrelenmiş bir den fazla veriye erişilmesi durumunda yapılan analizler sonucunda anahtarın tamamı ya da bir kısmı ele geçirilebilir. Bu da verinin tamamının ya da bir kısmının ele geçirilmesi demektir. Bu amaçla piksel değişim oranı sayısı (Number of Pixel Change Rate) ve birleşik ortalama değişken yoğunluğu (Unified Avarage Changing Intensity) değerlerinin yüzdelik olarak hesaplanması görüntü şifreleme sisteminin bu saldırılara karşı ne kadar dirençli olabileceğini gösterecektir. Değerlerin hesaplanma şekilleri 4.22, 4.23 ve 4.24'deki denklemlerde gösterilmiştir [212].

$$\Delta(i,j)=\begin{Bmatrix}0, & if\ P(i,j)=T(i,j)\\ (1, & if\ P(i,j)=T(i,j))\end{Bmatrix} \tag{4.22}$$

$$NPCR:N(P,T)=\sum_{i,j}\frac{\Delta(i,j)}{T}\times 100\% \tag{4.23}$$

$$UACI:U(P,T)=\sum_{i,j}\frac{|P(i,j)-T(i,j)|}{255\times MN}\times 100\% \tag{4.24}$$

NPCR ve UACI testleri, görüntü şifreleme sistemleri için diferansiyel saldırılara karşı direnci belirleme de en önemli parametrelerden birisidir. Test yapılırken görüntü şifreleme işlemi normal şekilde gerçekleştirilir. Ardından orijinal görüntünün bir pikseli değiştirilerek şifreleme işlemi aynı parametrelerle (aynı şifreleme anahtarı, aynı başlangıç değerleri gibi) tekrar yapılarak ikinci bir şifrelenmiş görüntü elde edilir. NPCR ve UACI testleri bu iki ayrı şifrelenmiş metin üzerinden yapılacaktır. Bu noktada en fazla yapılan hata, şifrelenmemiş görüntü ile şifrelenmiş görüntülerin NPCR ve UACI testlerinin yapılmasıdır. İki birbirinden farklı görüntünün NPCR ve UACI testleri yapılırsa sonuç doğal olarak başarılı olacaktır. Sadece bir piksel değişimine karşılık bu testlerin gerçekleştirilmesi sistemin gerçek manada güvenilir olup olmadığını belirleyecek önemli bir parametre olacaktır. Bu testler sonucunda NPCR oranı %100, UACI oranı ise %33'e yakın olarak hesaplandığında testlerin başarılı olduğu söylenebilir. Modern şifreleme sistemlerinin bu standardı sağlaması güvenilirliklerini kanıtlayacaktır.

4.5.6. Anahtar Hassasiyeti Analizi

Anahtar hassasiyeti analizi, şifreleme anahtarı içerisindeki bit değişimlerinin sistem güvenliğine etkisi olup olmadığını araştıran bir analizdir. Bu yöntemle, kriptografik sistem içerisinde farklı şifreleme anahtarlarının kullanılmasının sistem performansına olan etkisinin yanı sıra şifreleme anahtarının güvenilir olup olmadığı da bit düzeyindeki işlemlerle gösterilmektedir. Örneğin, bit düzeyindeki şifreleme anahtarının tamamı ya da büyük bir kısmı lojik-1 ifadelerinden oluşuyorsa bu anahtar tahmin edilebilir olarak sınıflandırılabilir. Şifreleme anahtarı üretim yöntemi olarak rastgele sayı üreteçlerinin kullanılması da tam olarak bu sebeptendir. Rastgele dağılımlı bir şifreleme anahtarının tahmin edilebilirliği güç olacağından daha iyi bir seçim olduğu gösterilebilecektir. Şifreleme anahtarı üzerinde 4.25'deki denklemde yer alan işlemler yapıldığında, anahtar hassasiyeti belirlenebilir.

$$T=\frac{n^a}{n^b}\% \tag{4.25}$$

Burada T değeri, bit dizisi içerisindeki lojik-0 ve lojik-1 ifadelerinin yüzdelik olarak dağılımını göstermektedir. n^a ve n^b değerleri ise şifreleme anahtarları içerisinde seçilen farklı bit dizilerinin lojik-0 ve lojik-1 oranlarını göstermektedir. Bu değerin ortalama olarak %50 olması seçilen şifreleme anahtarlarının ideal olduğunu gösterecektir [54]. Zaten, seçilen şifreleme anahtarları bir RSÜ çıkışı olarak alınmışsa bu durumda şifreleme anahtarlarının ideal olduğu belirlenebilir.

4.5.7. Ortalama Kare Hatası ve Tepe Sinyal Gürültü Oranı

Ortalama Kare Hatası, orijinal adıyla Mean-Squared-Error (MSE), genellikle bir sistem içerisindeki hata payını ölçmek için kullanılan önemli bir metriktir. Bu değerin "0"a yakın olarak elde edilmesi sistemin hata payının düşük olduğunun ve eğer bir tahmin yapılıyor ise sistemin elde ettiği değerlerin tutarlı olduğunun bir kanıtı olarak önerilen tahmin sisteminin çıktılarını destekleyebilir.

Tepe Sinyal Gürültü Oranı, orijinal adıyla Peak Signal to Noise Ratio (PSNR), en yüksek değerli sinyalin gürültüye olan oranını gösteren bir metriktir. MSE değeriyle beraber hesaplanarak genellikle iletişim sistemlerinin etkinlik düzeyini belirlemek için kullanılır. Şifreleme sistemleri içerisindeyse MSE ve PSNR değerleri tam tersi yönde bir uygulama alanı bulmuştur. Denklem 4.26 ve 4.27'deki formüller yardımıyla hesaplanabilirler [6].

$$MSE = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \left\| I(i.j) - C(i,j) \right\|^2 \tag{4.26}$$

$$PSNR = 20 \times \log_{10}\left(\frac{255}{\sqrt{MSE}}\right) dB \tag{4.27}$$

Önerilen şifreleme sistemleri için MSE değerleri yüksek ve PSNR değerleri düşük olarak hesaplandığında, hata payının yüksek olduğunu yani şifreleme sisteminin iyi çalıştığı sonucuna varılabilir. Genellikle görüntü şifreleme sistemleri için yapılan bu analizler literatürle karşılaştırılmalı olarak sonuçlar bölümünde sunulduğunda daha anlamlı olacaktır. Çünkü bu değerler için belirli bir aralık belirtmek mümkün değildir. Literatürdeki diğer başarılı kabul edilen şifreleme sistemlerinin MSE ve PSNR değerleriyle karşılaştırma yapılarak önerilen sistemin değerlendirilmesi sağlanacaktır.

4.5.8. Gürültü Saldırısı Analizi

Şifreleme işlemleri, gizli bir şekilde iletilmek istenilen verinin belirli bir haberleşme ortamında iletilerek güvenli bir şekilde alıcıya ulaştırılması amacıyla yapılmaktadır. Bu işlemler

sırasında istenilmeyen bir durum olsa da iletilen verinin bozulması muhtemeldir. Bu durum bazen bilgiyi elde etmek isteyen kişilerin iletişim kanalına isteyerek ya da bilgi sızdırma girişimleri sırasında istemeyerek gürültü eklemeleriyle sonuçlanabilir. Herhangi bir saldırı olmadığı durumlarda da haberleşme sisteminin kendisinden kaynaklanan sebeplerle iletilen veride bozulmalar meydana gelebilir. Yani şifrelenmiş bir veri iletim kanalında yolculuk ederken bir kısmı değişmiş olabilir.

Bu durum sonucunda bir kısım veri ister istemez kaybolsa da, elde edilen sınırlı veriyle orijinal veriye olabildiğince yakın şifre çözme işlemleri yapabilmek önemli bir parametre olacaktır. Örneğin, büyük bir veri iletimi sırasında çok küçük bir veri kaybının olması tüm bilginin yok olmasına yol açıyorsa bu şifreleme sisteminin kullanım alanı çok sınırlı kalabilir.

Şifreleme sistemleri için bu alanda çeşitli standartlar araştırmacılar tarafından belirlenmiştir. Genellikle görüntü şifreleme sistemleri içerisinde 0,005 yani binde 5 oranında şifrelenmiş veriye gürültü eklendiğinde şifre çözücü orijinal görüntüyü ufak gürültülerle elde edebiliyorsa şifreleme sistemleri bu tarz saldırılara karşı dirençli kabul edilir [213].

Gerçekleştirilen tez çalışması kapsamında önerilen şifreleme sistemi üzerinde, şifrelenmiş veriye farklı yoğunluk oranlarında gürültü sinyali eklenerek şifre çözme işlemi çıktıları alınacaktır. Orijinal verilere yakın sonuçlar elde edilip edilemediği görüntü şifreleme uygulamasıyla gösterilmeye çalışılacaktır.

5. BULGULAR VE TARTIŞMA

Bu bölümde önerilen yöntemler ve literatürde yer alan yöntemler yardımıyla gerçekleştirilen uygulamalardan bahsedilmiştir. Öncelikle geliştirilen farklı GRSÜ ve SRSÜ tasarımlarından bahsedilecektir. Sonra elde edilen rastgele sayı dizilerinin farklı testlere göre başarılı sayılıp sayılamayacağı tartışılacaktır.

Elde edilen rastgele sayı üreteci çıktılarının mevcut durumdaki standart şifreleme algoritmalarında ve tez kapsamında geliştirilen şifreleme algoritması içerisinde ne şekilde uygulanabileceği irdelenecektir. Bu çalışmada önerilen şifreleme yönteminin, literatürdeki sistemlere göre üstünlükleri de gösterilecektir.

Geliştirilen şifreleme yönteminin önceki bölümde gösterilen tüm kriptanaliz yöntemlerine göre başarılı olduğu ispatlanacaktır. Tüm güvenlik gereksinimlerini sağlayan simetrik blok şifreleme algoritmasının uygulamaları bu bölüm altında ayrı ayrı incelenecektir.

Şifreleme uygulamaları genellikle görüntü üzerine yoğunlaşmıştır. Zaten önceki bölümde bahsedilen bazı kriptanaliz yöntemleri de görüntü şifreleme uygulamalarına özel yöntemlerdir. Literatüre bakıldığında da görüntü şifreleme uygulamaları üzerine büyük bir yoğunluk olduğu görülebilir. Bunun bazı temel sebepleri vardır. İlk sebebi, günümüzde görüntü iletim trafiğinin diğer veri türlerine göre çok daha fazla artmasıdır. Artık her bireyin elinde iyi düzeyde fotoğraf çekebilen mobil cihazların bulunması görüntü verilerinin çığ gibi büyümesine yol açmıştır. Tıbbi görüntüleme gibi özel alanlarda da veri boyutları her geçen gün yükselmektedir. Bir diğer sebep, yine gelişmiş uydu sistemleri ya da askeri düzenekler genellikle kritik öneme sahip görüntü verisine ihtiyaç duymaktadır. Stratejik değeri yüksek uydu fotoğrafları ya da hava araçları tarafından elde edilen görüntülerin güvenilir bir şekilde istenilen noktalara iletilmesi önemli bir parametredir.

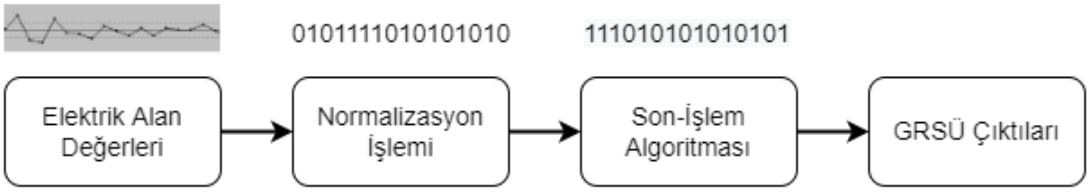
Son olarak bu tezde geliştirilen sistemde de olduğu gibi tasarlanan şifreleme sistemi her türlü veri için şifreleme yapabilse de görüntü şifreleme uygulamalarıyla başarımının daha iyi belirlenebilir olmasıdır. Görsel olarak orijinal görüntüyü ve şifrelenmiş görüntüyü incelemek sistem başarısını kanıtlamayı oldukça kolay hale getirmektedir. Rastgele bir veri kümesi şifrelendiğinde zaten başlangıçta da o veriyi nasıl kullanacağını bilmeyen kullanıcılar için şifrelenmiş veriyle orijinal veri arasında fark bulmak oldukça zor olabilir. Ayrıca NPCR ve UACI gibi görüntü şifreleme uygulamalarına özel olarak geliştirilen güvenlik testlerini rastgele bir veriye uygulamak mümkün değildir. Tüm bu yönleriyle görüntü şifreleme uygulamaları hem popülaritesinin günden güne arttırmakta hem de analiz sonuçlarını rahat gösterebilmesi sebebiyle araştırmacılara büyük kolaylıklar sağlamaktadır.

5.1. Rastgele Sayı Üreteci Uygulamalarının Değerlendirilmesi

Bu bölümde geliştirilen rastgele sayı üreteçlerinin analizleri yapılmıştır. Geliştirilen GRSÜ ve SRSÜ tasarımları istatistiksel analizler yardımıyla değerlendirilmiştir. Başarılı olunan ve başarısız olunan testlerle ilgili tartışmalar yapılmıştır.

5.1.1. Ölçüm Verileriyle Oluşturulan GRSÜ Analizi

Geliştirilen rastgele sayı üreteçleri farklı analizler yardımıyla değerlendirilmeye çalışılmıştır. İlk olarak elektrik alan ölçüm verileriyle daha önce uygulanmamış bir GRSÜ tasarlanmıştır. Elde edilen veriler ham olarak normalize edilmiş GRSÜ ve XOR son işlem algoritması uygulanmış GRSÜ olarak iki farklı yapıda değerlendirilmiştir. Genel olarak Elektrik Alan verileriyle GRSÜ tasarlamak için gerekli olan adımlar Şekil 5.1'de gösterilmiştir.



Şekil 5.1. Elektrik alan verileriyle GRSÜ tasarlamak için gerekli temel adımlar

Bu adımlar doğrultusunda gerçekleştirilen GRSÜ çıktıları öncelikle istatistiksel testlerle değerlendirilmiştir. Hem normalizasyon adımı sonrasında hem de son-işlem adımı sonrasında elde edilen rastgele bit dizileri istatistiksel olarak değerlendirilmiştir. Buna göre ilk olarak NIST testlerinde elde edilen sonuçlar Tablo 5.1'de verilmiştir.

Tablo 5.1. Ölçüm değerleriyle oluşturulan GRSÜ NIST testi sonuçları

Test Adı	Normalize Edilmiş Değerler		Son İşlem Uygulanan Değerler	
	Başarım	**P-Değeri**	**Başarım**	**P-Değeri**
Frekans Testi	Başarılı	0.77	Başarılı	0.72
Bir Blok içerisinde Frekans Testi	Başarılı	0.75	Başarılı	0.71
Akış Testi	Başarılı	0.72	Başarılı	0.35
Bir Blok içerisinde en Uzun Akış Testi	Başarılı	0.56	Başarılı	0.28
İkili Matris Derece Testi	Başarılı	0.02	Başarılı	0.02
Ayrık Zamanlık Fourier Dönüşüm Testi	Başarılı	0.41	Başarılı	0.46
Örtüşmeyen Şablon Testi	Başarılı	0.13	Başarılı	0.29
Örtüşen Şablon Testi	Başarılı	0.36	Başarılı	0.60
Evrensel Test (Maurer Testi)	Başarılı	0.63	Başarılı	0.41
Doğrusal Karmaşıklık Testi	Başarılı	0.02	Başarılı	0.29
Seri Testi	Başarılı	0.36/0.39	Başarılı	0.45/0.74
Yaklaşık Entropi Testi	Başarılı	0.39	Başarılı	0.20
Birikimli Toplamlar Testi	Başarılı	0.50	Başarılı	0.35
Rastgele Gezinimler Testi	Başarılı	0.45	Başarılı	0.36
Rastgele Gezinimler Değişken Testi	Başarılı	0.18	Başarılı	0.24

Hem normalize edilen değerler hem de son işlem uygulanan değerler NIST testlerinden başarılı sonuçlar almıştır. Bir başka analiz olan Otokorelasyon testiyle ilgili sonuçlar Tablo 5.2'de paylaşılmıştır.

Tablo 5.2. Ölçüm değerleriyle oluşturulan GRSÜ Otokorelasyon test sonuçları

Veri	$\lvert X5 \rvert$ değerleri							
	4	**8**	**10**	**25**	**35**	**80**	**125**	**250**
Normalize Edilmiş Değerler	5.8870	5.1442	2.6257	5.4403	2.4888	3.6393	2.1555	2.5554
Son İşlem Uygulanan Değerler	0.0509	0.7032	0.2884	1.4818	1.0369	0.5826	1.4594	0.1301

Otokorelasyon testine göre elde edilen $|X5|$ değerleri 1,6449'dan küçük olmalıdır. Bu teste göre son işlem uygulanan değerlerin başarılı olduğu sadece normalizasyon uygulanan değerlerin ise başarılı olamadığını görüyoruz. Böylece tasarlanan rastgele sayı üreteçlerinin genel manasıyla başarılı görünseler bile son işlem algoritmalarıyla istatistiksel özelliklerinin geliştirilebileceğini görüyoruz.

Bir diğer RSÜ testi olan Scale-Index testine ilişkin sonuçlar Tablo 5.3'te gösterilmektedir. Buna göre iki veri grubu da bu testten başarım sağlamıştır.

Tablo 5.3. Ölçüm değerleriyle oluşturulan GRSÜ Scale-Index test sonuçları

Veri	N*[10000]								
	1-2	**2-3**	**3-4**	**4-5**	**5-6**	**6-7**	**7-8**	**8-9**	**9-10**
Normalize Edilmiş Değerler	0.8537	1	0.9517	0.8601	0.8114	0.8509	0.9756	0.8701	0.9494
Son İşlem Uygulanan Değerler	0.8599	1	0.9233	0.9383	0.9645	0.9321	1	0.9252	0.7799

Scale-Index testine göre değerlerin "1"e yakın elde edilmesi sonuçların başarılı olduğunu gösterecektir. Buna göre her iki bit dizisi de başarılı sonuçlar elde etmiştir. Tablo 5.4'te her iki bit dizisine ait "1" ve "0" değerlerinin oranları verilmiştir.

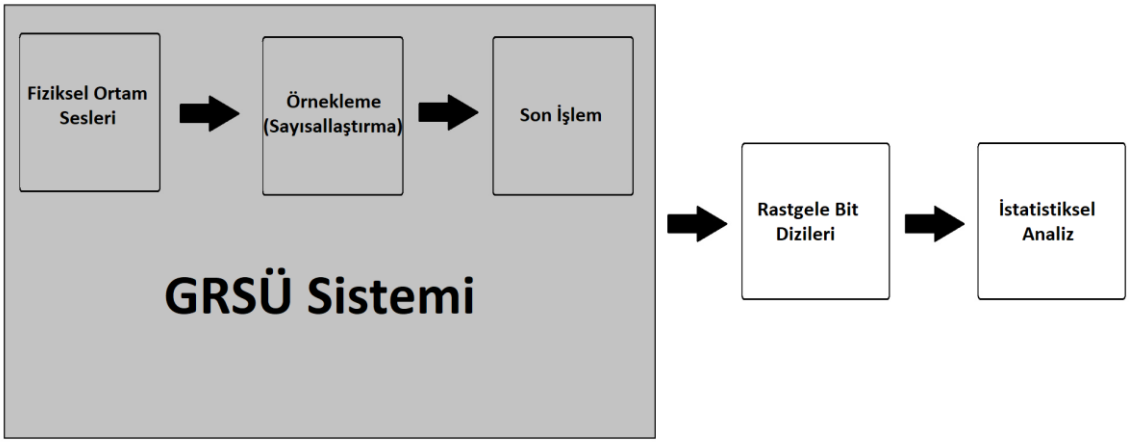
Tablo 5.4. Ölçüm değerleriyle oluşturulan GRSÜ bit oranları

Veri	Bit Oranı
Normalize Edilmiş Değerler	0.49904
Son İşlem Uygulanan Değerler	0.50009

Buna göre her iki veri grubunda da yaklaşık olarak lojik-1 ve lojik-0 ifadelerinin eşit sayıda bulunduğu söylenebilir. Bu da veri gruplarının rastgele bir dağılıma sahip olduğunu herhangi bir değerin yoğunlaşmadığını göstermektedir.

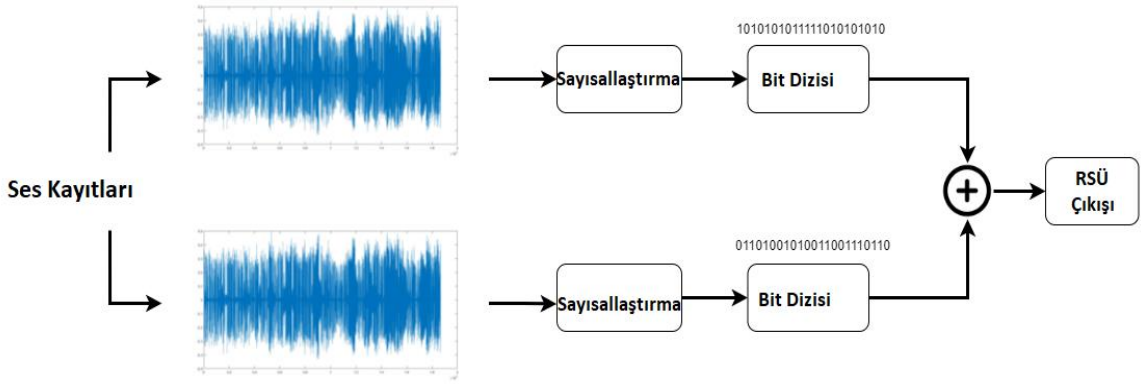
5.1.2. Ses Verileri Yardımıyla GRSÜ Tasarımı

Ses verileriyle GRSÜ tasarımı yapılabileceğini göstermek için iki farklı yöntem izlenmiştir. İlk olarak literatürde hazır bulunan ESC-50 ve UrbanSound8K gibi çevresel ses veri setlerinde bulunan ses dosyaları yardımıyla GRSÜ tasarlanmıştır. İkinci olarak ise uygun bir ortamda çevresel ses kaydı alınıp kendi oluşturduğumuz veri grubu yardımıyla GRSÜ tasarımı yapılarak aradaki farklılıklar incelenmiştir. Geliştirilen sistemin genel şeması Şekil 5.2'de verilmiştir.



Şekil 5.2. Ses verileriyle GRSÜ tasarımı blok şeması

Blok şemasına göre fiziksel ortamdan kayıt yoluyla ya da veri setinden alınarak elde edilen ses dosyaları sayısallaştırılarak GRSÜ sisteminde kullanılmaktadır. Son işlem sonrası elde edilen rastgele bit dizisi istatistiksel analizlerle değerlendirilmiştir. Şekil 5.3'de ise rastgele sayı üreteci katmanları gösterilmiştir.



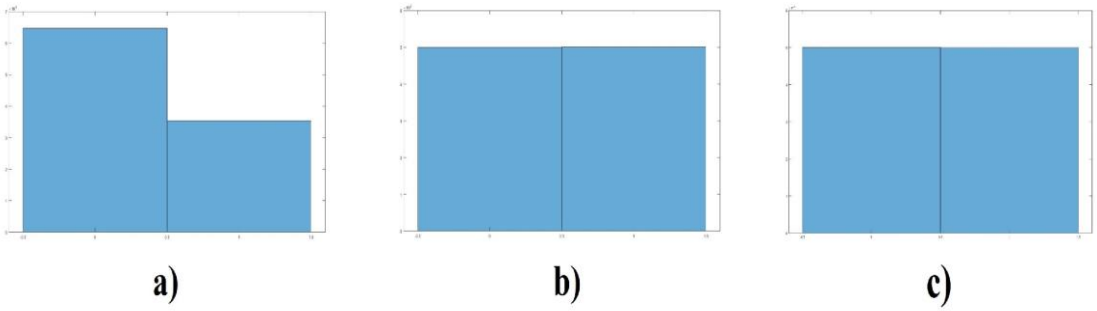
Şekil 5.3. Ses kayıtlarının RSÜ tasarımında kullanımı

Burada RSÜ tasarımı yapılırken son işlem algoritması olarak XOR lojik operasyonunun kullanıldığı görülmektedir. Tablo 5.5'de geliştirilen GRSÜ tasarımlarına ait NIST test sonuçları paylaşılmıştır.

Tablo 5.5. Ses verileriyle oluşturulan GRSÜ NIST testi sonuçları

Test No	Orijinal Veri Seti		Son İşlem Uygulanan Veri Seti		Ortam Ses Kayıtları	
	Başarım	**P-Değeri**	**Başarım**	**P-Değeri**	**Başarım**	**P-Değeri**
1	Başarısız	0	Başarılı	0.43	Başarılı	0.76
2	Başarısız	0	Başarılı	0.72	Başarılı	0.39
3	Başarısız	0	Başarılı	0.31	Başarılı	0.98
4	Başarısız	0	Başarılı	0.99	Başarılı	0.13
5	Başarısız	0	Başarılı	0.51	Başarılı	0.74
6	Başarısız	0	Başarılı	0.34	Başarılı	0.56
7	Başarısız	0	Başarılı	0.02	Başarılı	0.04
8	Başarısız	0	Başarılı	0.14	Başarılı	0.20
9	Başarısız	0	Başarılı	0.79	Başarılı	0.94
10	Başarısız	0	Başarılı	0.75	Başarılı	0.42
11	Başarısız	0 / 0	Başarılı	0.39/0.34	Başarılı	0.97/0.96
12	Başarısız	0	Başarılı	0.39	Başarılı	0.65
13	Başarısız	0	Başarılı	0.74	Başarılı	0.66
14	Başarısız	0	Başarılı	0.65	Başarılı	0.53
15	Başarısız	0	Başarılı	0.58	Başarılı	0.51

Burada orijinal ses verisi dosyalarıyla oluşturulan bit dizilerinin hiç bir testte başarı sağlayamadığı yani rastgele sayı dizisi olarak kullanılamayacağı görülmektedir. Ancak son işlem uygulanan veri grubunun tüm testlerde başarı sağladığı belirlenmiştir. Buna göre ses dosyalarına son işlem uygulamasının gerekli olduğunu söyleyebiliriz. Ayrıca veri setinden alınan örneklerle ortam seslerinin kaydedilmesi arasında da herhangi bir fark bulunmadığı söylenebilir. Orijinal ses ve GRSÜ çıkışındaki ses verisinin ikili sayı düzeninde histogram grafikleri Şekil 5.4'te verilmiştir.



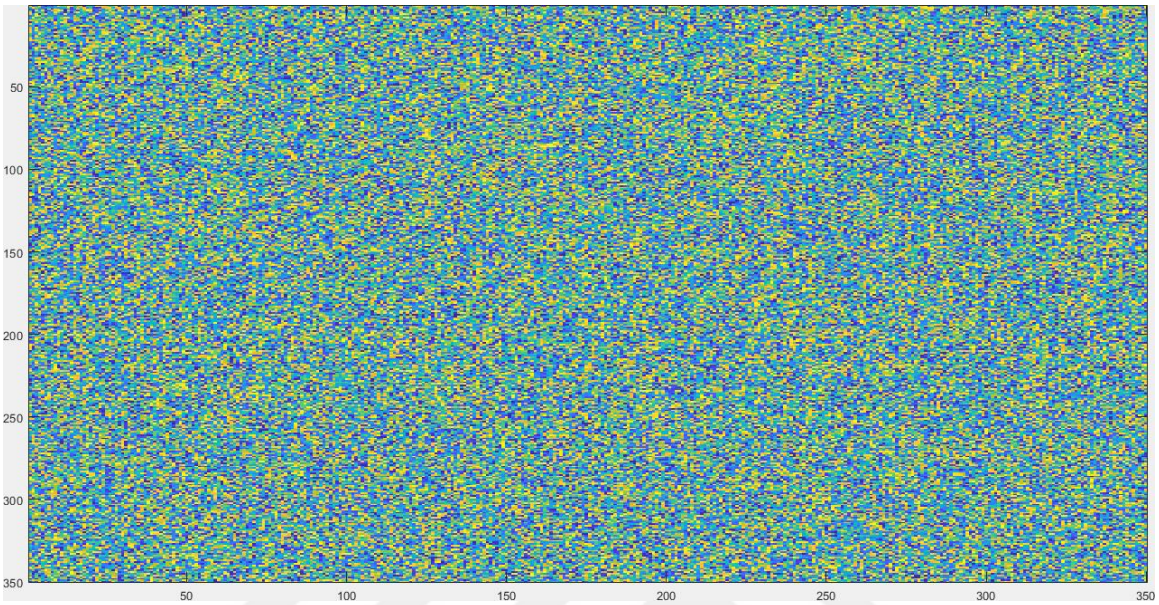
Şekil 5.4. Ses verileri histogramı **a)** orijinal veri örneği **b)** veri setinden üretilen GRSÜ **c)** kaydedilen sesten üretilen GRSÜ

Histogram analizine bakıldığında orijinal verilerin bit dağılımında bariz bir şekilde dengesizlik olduğu söylenebilir. Bit dağılımı bu düzeyde dengesiz olduğunda herhangi bir işlem yapılmadan GRSÜ tasarlanması mümkün değildir. Geliştirilen bu rastgele sayı üreteci sisteminin literatürdeki benzer çalışmalarla karşılaştırması Tablo 5.6'da gösterilmiştir.

Tablo 5.6. Ses verileriyle oluşturulan GRSÜ literatür karşılaştırması

Ref.	Üretici Çeşidi	Gerçekleştirme Türü	Rastgelelik Kaynağı	Son İşlem	Üretim Hızı	Testler
[50]	SRSÜ	Simulation	LCG	Trivium		NIST
[23]	SRSÜ	Simulation	Kaotik Harita	-	-	NIST, Scale-Index, Otokorelasyon
[19]	GRSÜ	Simulation	Güç Jeneratörü	-	-	NIST
[214]	GRSÜ	Simulation	Ses Sinyali	-	-	NIST, TestU01
[22]	GRSÜ	FPGA	RO	XOR	95,37 Mbit/s	NIST
[34]	GRSÜ	Analog Circuit	RO	XOR	3,02 Mbit/s	NIST
[45]	SRSÜ	Simulation	LFSR, Kaotik Harita	-	14,48 Mbit/s	NIST, DIEHARD
[25]	GRSÜ	Simulation	Memristor	Trivium	-	NIST, Scale-Index
[66]	GRSÜ	Simulation	Elektromanyetik Kirlilik	XOR	-	NIST, Scale-Index, Otokorelasyon
[47]	SRSÜ	Simulation	MARC-bb Algoritması	-	131,15 Kbit/s	TestU01
Önerilen	GRSÜ	Simulation	Ortam Sesleri	XOR	48,06 Mbit/s	NIST

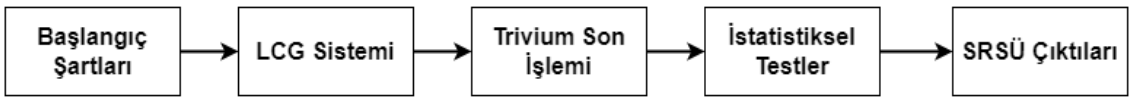
Üretilen rastgele bit dizisinin 8-bit görüntü formatına çevrilmiş renkli görüntüsü Şekil 5.5'te gösterilmiştir. Elde edilen görüntü de üretilen bit dizisinin rastgele formda olduğunu destekler niteliktedir.



Şekil 5.5. Ses verisiyle gerçekleştirilen GRSÜ'nün 8-bit görüntüsü

5.2. LCG ve Trivium Yardımıyla SRSÜ Tasarımı

Doğrusal Eşlenik Üreteci (LCG) ve Trivium algoritması yardımıyla istatistiksel açıdan iyi sonuçlar elde eden bir SRSÜ tasarımı gerçekleştirilmiştir. Sistemin genel yapısı Şekil 5.6'da gösterilmektedir.



Şekil 5.6. LCG-Trivium SRSÜ blok şeması

Burada LCG üretimi için belirli başlangıç parametreleri belirlenerek bit üretimine başlanır. Bu parametreler bit üretim sistemini yakından etkilese de son işlem algoritmalarının kullanılacağı sistemlerde çok da önemli olmayacaktır. Güçlü bir rastgelelik algoritması olan Trivium üretilen bitler rastgele olmasa bile kaliteli bir SRSÜ çıktısı sağlayabilmektedir. Burada alınan çıktılar istatistiksel testlerle değerlendirilerek geliştirilen SRSÜ'nün başarım oranı belirlenecektir. Tablo 5.7'de LCG-Trivium SRSÜ'nün NIST test sonuçları verilmiştir.

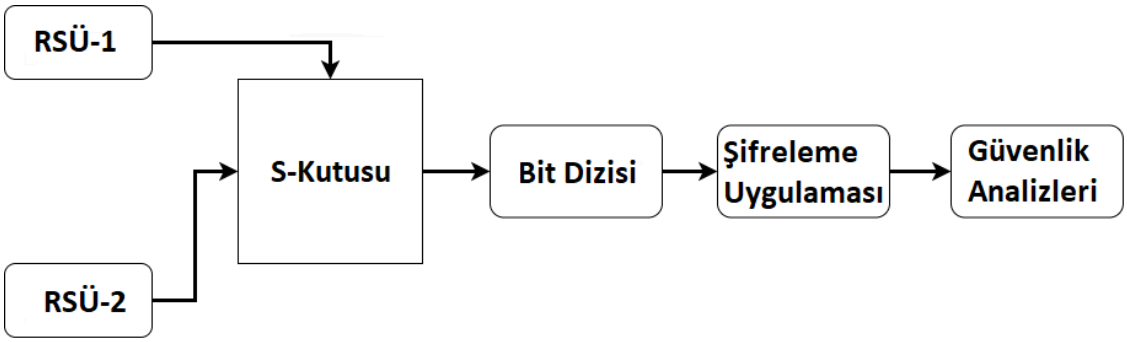
Tablo 5.7. LCG-Trivium SRSÜ NIST testi sonuçları

Test Adı	LCG Çıktıları		Trivium Son İşlemi Uygulanan Değerler	
	Başarım	**P-Değeri**	**Başarım**	**P-Değeri**
Frekans Testi	Başarısız	0	Başarılı	0.79
Bir Blok içerisinde Frekans Testi	Başarısız	0	Başarılı	0.27
Akış Testi	Başarısız	0	Başarılı	0.25
Bir Blok içerisinde en Uzun Akış Testi	Başarısız	0	Başarılı	0.34
İkili Matris Derece Testi	Başarılı	0.21	Başarılı	0.07
Ayrık Zamanlık Fourier Dönüşüm Testi	Başarısız	0	Başarılı	0.41
Örtüşmeyen Şablon Testi	Başarısız	0	Başarılı	0.03
Örtüşen Şablon Testi	Başarısız	0	Başarılı	0.32
Evrensel Test (Maurer Testi)	Başarısız	0	Başarılı	0.20
Doğrusal Karmaşıklık Testi	Başarısız	0	Başarılı	0.91
Seri Testi	Başarısız	0/0	Başarılı	0.95/0.94
Yaklaşık Entropi Testi	Başarısız	0	Başarılı	0.13
Birikimli Toplamlar Testi	Başarısız	0	Başarılı	0.54
Rastgele Gezinimler Testi	Başarısız	0	Başarılı	0.30
Rastgele Gezinimler Değişken Testi	Başarısız	0	Başarılı	0.31

NIST testlerinin sonuçlarına göre LCG çıktıları sadece bir testte başarılı olurken, Trivium son işlemi uygulanan bit dizileri tüm testlerden başarılı olmuştur. Bu durumda LCG çıktılarının doğrudan SRSÜ olarak kullanımı uygun değilken Trivium son işlemi uygulandığında başarılı bir SRSÜ elde edilmiş olur.

5.3. S-Kutusu Destekli Rastgele Sayı Üreteci Tasarımı

S-Kutularının rastgele sayı üreteci tasarımları içerisinde kullanılabileceğinden ya da bir son işlem algoritması olarak kullanılabileceği daha önceki bölümler de açıklanmıştı. Burada Şekil 5.7'de gösterilen şekilde bir RSÜ geliştirilmiştir.



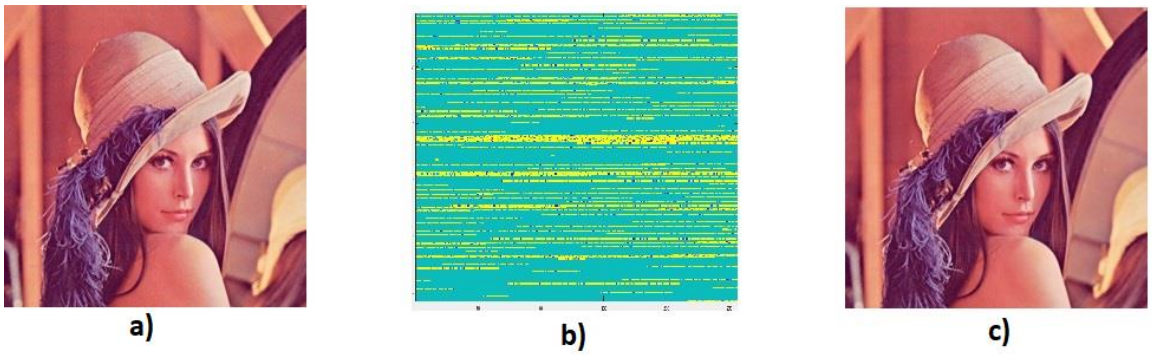
Şekil 5.7. S-Kutusu tabanlı şifreleme sistemi şeması

Bu şekilde geliştirilen iki farklı rastgele sayı üreteci S-Kutusu yardımıyla daha farklı bir rastgele sayı üretecine dönüşmektedir. Tabi ki bu sistemin daha farklı tasarımları da gerçekleştirilebilir. Gerçekleştirilen tasarımda RSÜ-1 sütun seçiminden sorumlu iken RSÜ-2 satır seçimi yaparak S-Kutusundan değerleri seçmektedir. Bu üç farklı sistemin NIST test sonuçları Tablo 5.8'de verilmiştir.

Tablo 5.8. S-Kutusu tabanlı RSÜ'lere ilişkin NIST testi sonuçları

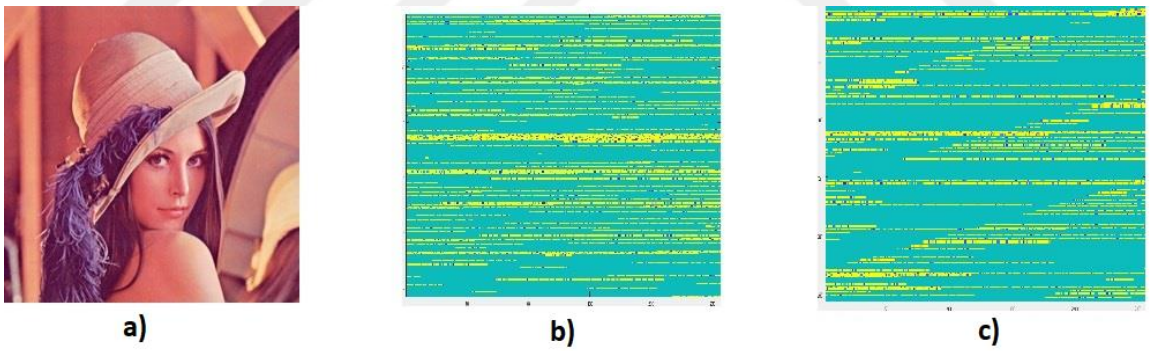
Test No	Keccak Algoritması		Farfalle Fonksiyonu		S-Kutusu Çıktısı	
	Başarım	P-Değeri	Başarım	P-Değeri	Başarım	P-Değeri
1	Başarılı	0.34	Başarılı	0.72	Başarılı	0.29
2	Başarılı	0.24	Başarılı	0.40	Başarılı	0.62
3	Başarılı	0.28	Başarılı	0.56	Başarılı	0.32
4	Başarılı	0.04	Başarılı	0.77	Başarılı	0.89
5	Başarılı	0.83	Başarılı	0.40	Başarılı	0.98
6	Başarılı	0.41	Başarılı	0.85	Başarılı	0.27
7	Başarılı	0.02	Başarılı	0.78	Başarılı	0.80
8	Başarılı	0.57	Başarılı	0.57	Başarılı	0.82
9	Başarılı	0.39	Başarılı	0.52	Başarılı	0.42
10	Başarılı	0.43	Başarılı	0.10	Başarılı	0.76
11	Başarılı	0.71/0.60	Başarılı	0.75/0.88	Başarılı	0.99/0.77
12	Başarılı	0.40	Başarılı	0.12	Başarılı	0.84
13	Başarılı	0.30	Başarılı	0.60	Başarılı	0.38
14	Başarılı	0.32	Başarılı	0.48	Başarılı	0.61
15	Başarılı	0.28	Başarılı	0.21	Başarılı	0.79

NIST testleri sonuçlarına göre tüm RSÜ tasarımlarının başarılı olduğu söylenebilir. Burada elde edilen bit dizileri AES şifreleme algoritmasında şifreleme anahtarı olarak kullanılmıştır örnek şifreleme uygulaması Şekil 5.8'de gösterilmiştir.



Şekil 5.8. S-Kutusu RSÜ ile AES şifreleme örneği **a)** orijinal görüntü **b)** şifrelenmiş görüntü **c)** şifresi çözülmüş görüntü

Buna göre AES ile şifreleme işlemi başarılı olarak gerçekleştirilmiştir. Anahtar hassasiyeti incelendiğinde şifre çözme işleminde anahtarın bir bit'i bile değiştirildiğinde elde edilen şifresi çözülmüş görüntü Şekil 5.9'da görüldüğü gibidir.



Şekil 5.9. S-Kutusu RSÜ ile bir bit değişimli anahtar ile şifre çözme **a)** orijinal görüntü **b)** şifrelenmiş görüntü **c)** şifresi çözülmüş görüntü

Görüldüğü gibi şifreleme anahtarında bir bit'lik bile değişim yapılırsa orijinal görüntüyle hiçbir alakası olmayan görüntü elde edilmiş olur. T şifreli metindeki "1" ve "0"ların oranları ise tüm yapılan denemelerin ortalamasında %49.908 olarak tespit edilmiştir. Bit oranlarının yaklaşık olarak aynı olması şifreleme sisteminin başarılı olduğunu göstermektedir. Şifrelenmiş görüntülere ilişkin NPCR ve UACI değerleri Tablo 5.9'da gösterilmiştir.

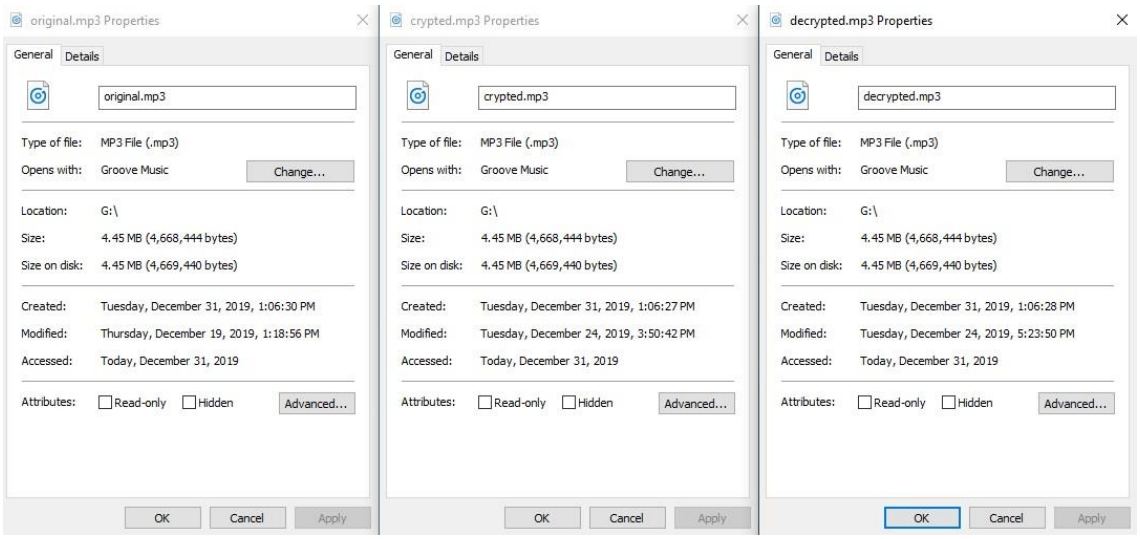
Tablo 5.9. S-Kutusu RSÜ ile AES şifreleme sisteminin NPCR ve UACI testleri

Test	Şifrelenmiş Metin	İdeal Değerler
NPCR	%99,909	%100
UACI	%33,337	%33

Görüldüğü gibi Vernam şifreleme sisteminin aksine AES şifreleme uygulaması yapıldığında NPCR ve UACI testlerinden başarılı sonuçlar alınmaktadır.

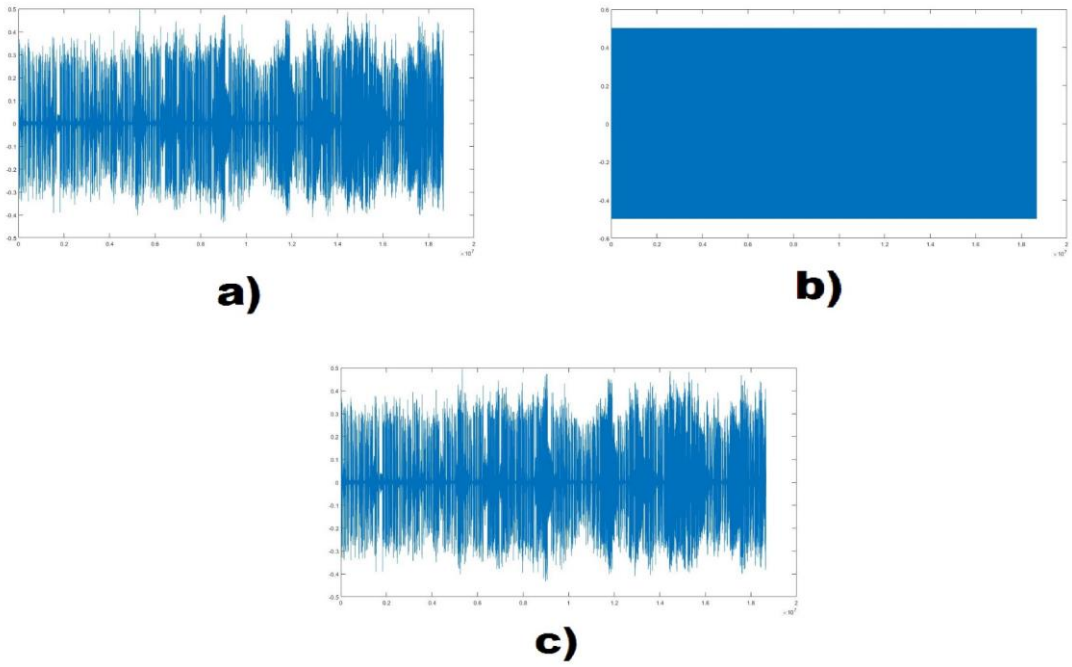
5.4. Ses Şifreleme Uygulaması

Yapılan tez çalışması kapsamında ses verilerinden rastgele sayı üreteci tasarlamıştık. Ses dosyalarını şifrelemek için de literatürde bulunan standart yöntemlerden bazılarını kullandık. Öncelikle standart bir ses dosyasını literatürdeki en eski yöntemlerden olan Vernam şifreleyicisi kullanarak şifreledik. Bu yöntemin bazı güvenlik açıkları olsa da rastgele sayı üreteci çıktıları şifreleme anahtarı olarak kullanıldığında iyi sonuçlar elde edebilmektedir ve çok hızlı çalışan bir yapıda olduğu için test etmek istedik. Şifreleme yapılan dosya Şekil 5.10'da gösterilmiştir. Burada herhangi bir veri kaybı olmadan şifreleme ve şifre çözme işlemlerinin gerçekleştirildiği görülmektedir.



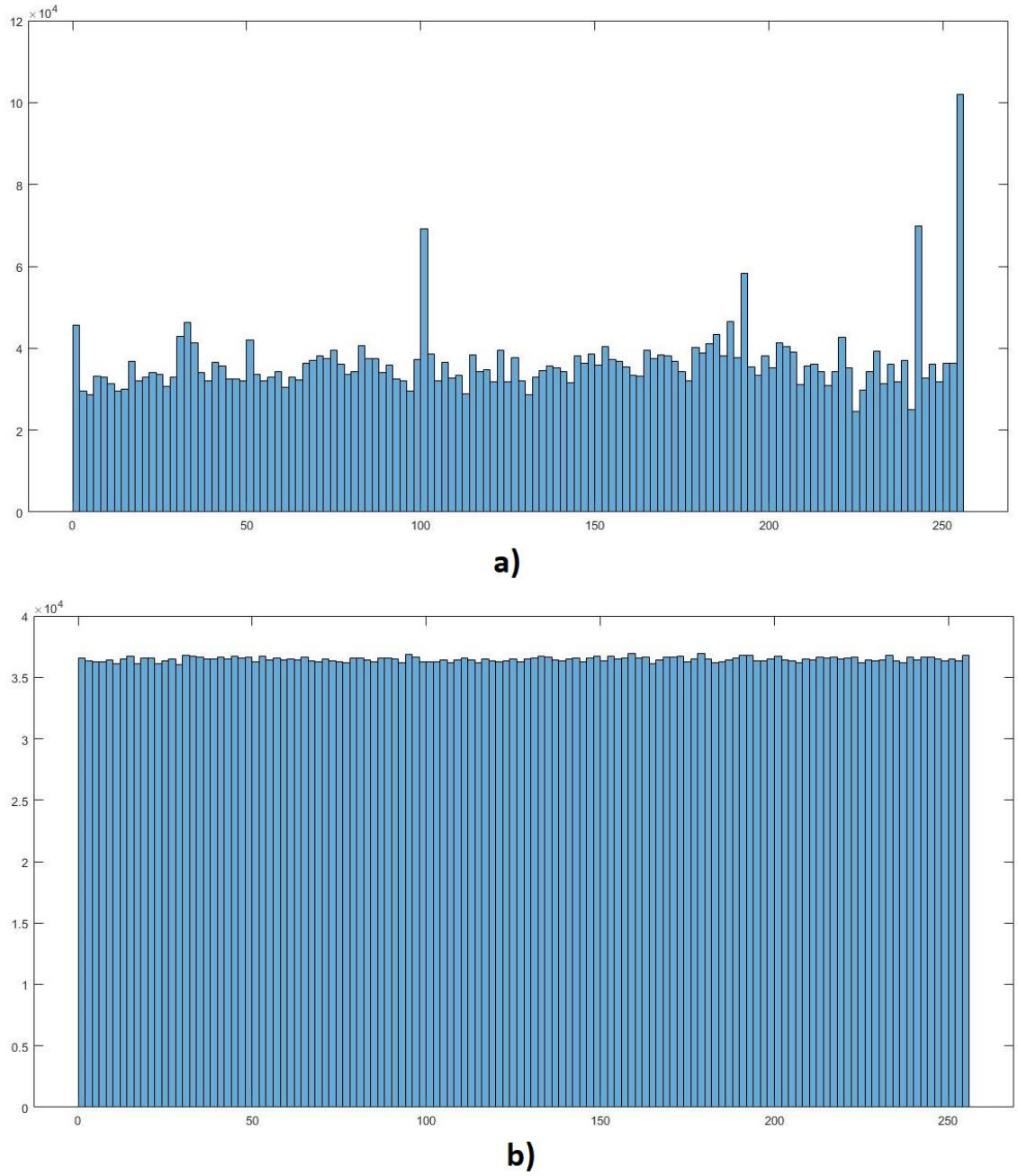
Şekil 5.10. Orijinal, şifrelenmiş ve şifresi çözülmüş örnek ses dosyası

Vernam şifreleyicisi birebir işlem yaptığı için şifrelenen ses dosyası tamamen orijinal dosya gibidir ancak Şekil 5.11'da görüleceği üzere ses dosyası anlaşılamaz bir yapıdadır.



Şekil 5.11. Ses şifreleme uygulaması: **a)** orijinal ses **b)** şifrelenmiş ses **c)** şifresi çözülmüş ses

Görüldüğü gibi şifrelenmiş ses dosyasında ses tonlarını tespit etmek ya da bir benzetim yapmak mümkün değildir. Orijinal sesin ve şifrelenmiş sesin desimal değerler üzerinden histogram analizi Şekil 5.12'de gösterilmiştir.



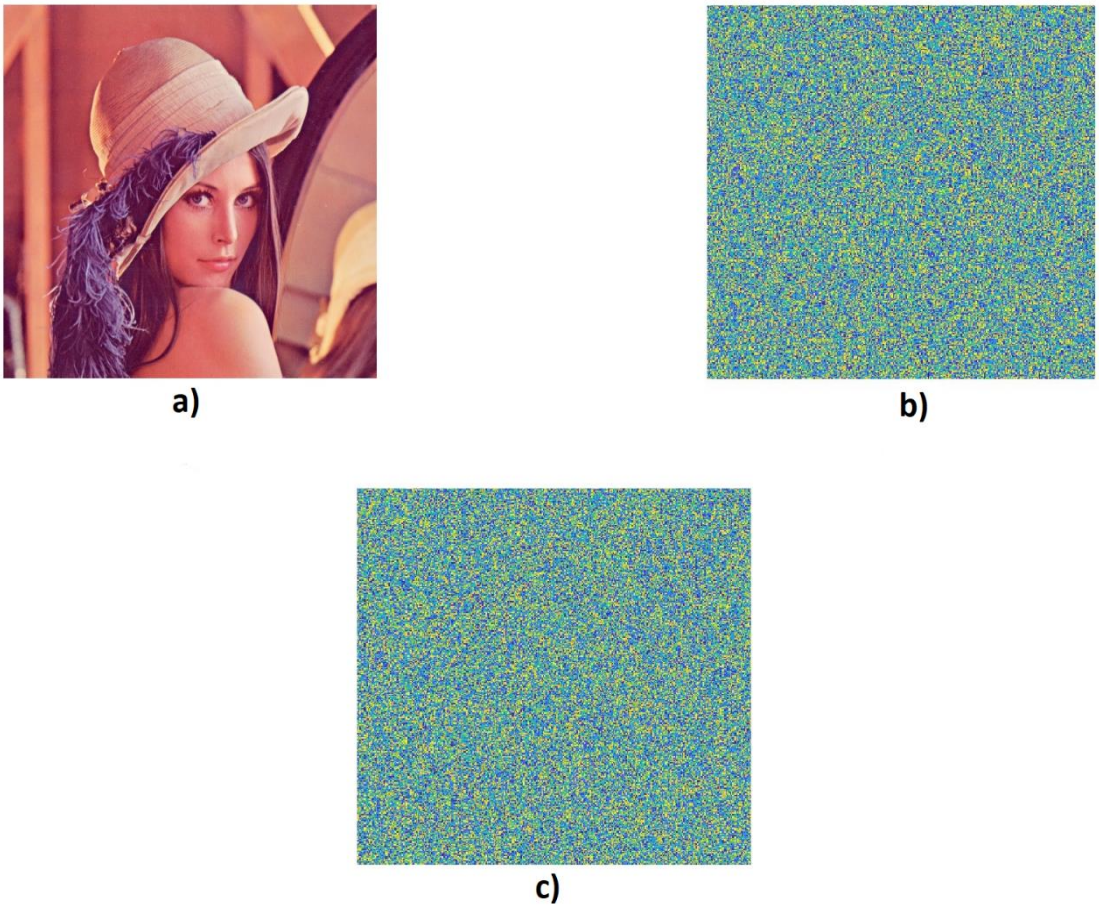
Şekil 5.12. Örnek ses şifreleme histogram analizi: **a)** orijinal ses **b)** şifrelenmiş ses

Görüldüğü gibi orijinal ses verisinde belirli yoğunlaşmalar bulunsa da şifrelenmiş ses verisi için tüm değerlerin sayısı yaklaşık olarak birbiriyle aynıdır. Bu da iyi bir şifreleme yapıldığını göstermektedir.

5.5. Ses Verisiyle Üretilen GRSÜ Yardımıyla Görüntü Şifreleme

Bu uygulama kapsamında ses verileriyle oluşturulan GRSÜ çıktıları AES şifreleme algoritmasına uygulanarak örnek uygulamalar gerçekleştirilmiştir. Ayrıca görüntü şifreleme

işlemleri için özel olarak geliştirilen NPCR ve UACI testleri ilk olarak görüntü şifreleme işlemlerine uygulanan AES uygulaması üzerinde test edilmiştir. Şekil 5.13'te örnek Lena görüntüsünün orijinalinin ve bir piksel değiştirilmiş versiyonunun şifrelenmiş görüntüleri gösterilmiştir.



Şekil 5.13. AES ile örnek görüntü şifreleme **a)** orijinal görüntü **b)** şifrelenmiş görüntü **c)** piksel değişimli şifrelenmiş görüntü

Görüldüğü gibi AES algoritması küçük bir piksel değişiminde bile farklı bir şifrelenmiş görüntü oluşturmayı başarmıştır. Bu durumun istatistiksel olarak kanıtlanması için Tablo 5.10'da gösterilen NPCR, UACI ve Entropi testleri uygulanarak literatürdeki sonuçlarla karşılaştırılmıştır.

Tablo 5.10. AES ile şifrelenen görüntüye ilişkin NPCR, UACI ve Entropi testi karşılaştırması

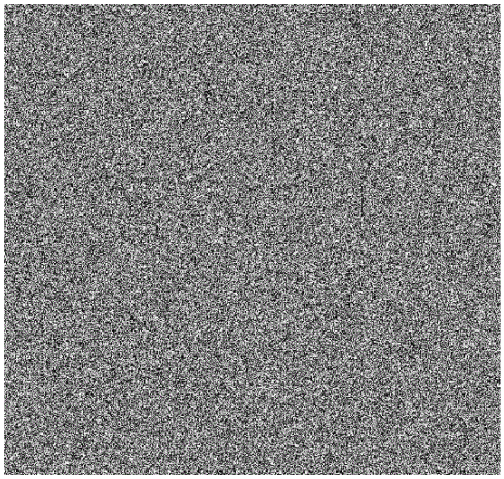
Referans	Number of Pixel Change Rate (NPCR)	Unified Average Changing Intensity (UACI)	Entropy Testi (Ent)
Önerilen	%99.6187	%33.4635	7.9984
[142]	%99.6100	%33.4515	7.9993
[159]	%99.6000	%33.5600	7.9973
[168]	%99.7024	%33.5246	7.9984
[215]	%99.6200	%33.8120	7.9975
[216]	%99.6067	%33.4951	7.9993
[217]	%99.6413	%33.4801	-
[218]	%99.6002	%33.5079	7.9994
[219]	%99.6552	%33.4846	-
[176]	%99.9999	%33.7400	7.9912
[220]	%99.9100	%33.3500	7.9978
[221]	%99.6143	%33.6532	7.9974
[222]	%99.6059	%33.4161	7.9972

Tabloya göre yapılan analizin literatürdeki değerlerle paralel olduğu belirlenmiştir. Buna göre gerçekleştirilen görüntü şifreleme uygulamasının başarılı olduğu söylenebilir.

5.6. Vernam Şifreleyicisiyle Görüntü Şifreleme Uygulaması

Yapılan bu uygulamada temel düzeyde bir şifreleme gerçekleştirmek için rastgele sayı üreteci çıktıları kullanılmıştır. Örnek gri tonlamalı görüntüler üzerinde yapılan uygulamalar Şekil 5.14 ve Şekil 5.15'te gösterilmiştir.

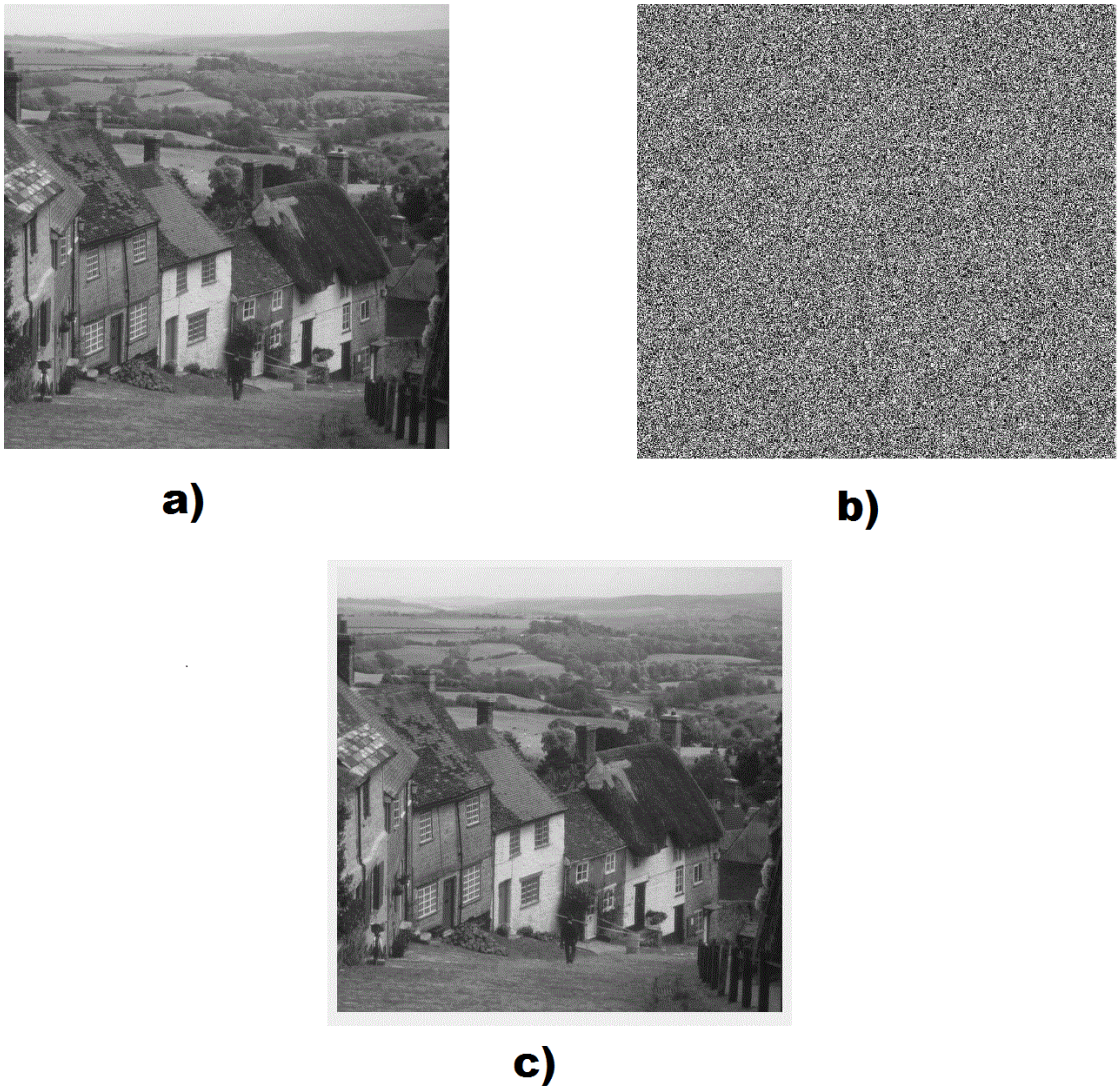
a)



b)

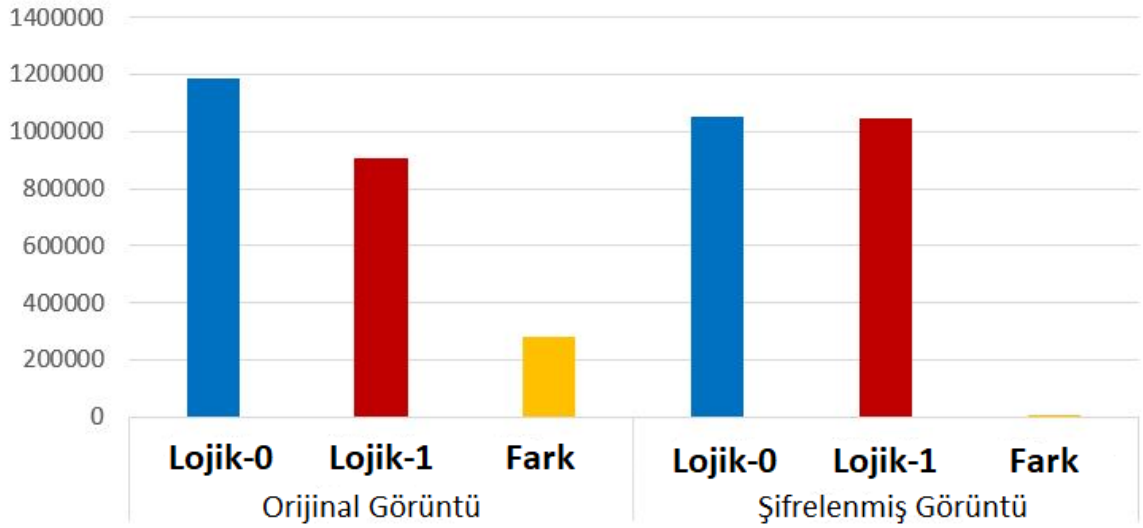
c)

Şekil 5.14. Vernam şifreleyiciyle örnek gri tonlamalı görüntü şifreleme1eme 1: **a)** orijinal görüntü **b)** şifrelenmiş görüntü **c)** şifresi çözülmüş görüntü



Şekil 5.15. Vernam şifreleyiciyle örnek gri tonlamalı görüntü şifreleme1eme 2: **a)** orijinal görüntü **b)** şifrelenmiş görüntü **c)** şifresi çözülmüş görüntü

Şifrelemeler gri tonlamalı görüntüler üzerine başarıyla uygulanmıştır. Örnek bir şifreleme uygulamasının bit dağılımları Şekil 5.16'da verilmiştir.



Şekil 5.16. Gri tonlamalı görüntü şifreleme uygulamasının bit dağılımları

Buna göre lojik-0 ve lojik-1 değerlerinin dağılımları orijinal görüntüde düzgün değilken şifrelenmiş görüntüde bit dağılımlarının düzgün olduğu, başarılı bir şifreleme geliştirildiği belirlenmiştir.

Vernam şifreleyicisi eski bir şifreleyici olduğu için modern güvenlik koşullarını sağlamamaktadır. Örneğin piksel değişimi yapılarak şifrelenme yapıldığında sadece ilgili pikselin şifrelenmiş görüntüsü değişmektedir. Diğer pikseller aynı kalmaktadır. Bu durumda NPCR ve UACI testleri bu şifreleyiciye uygulandığında sonuçların başarısız olduğu görülmüştür.

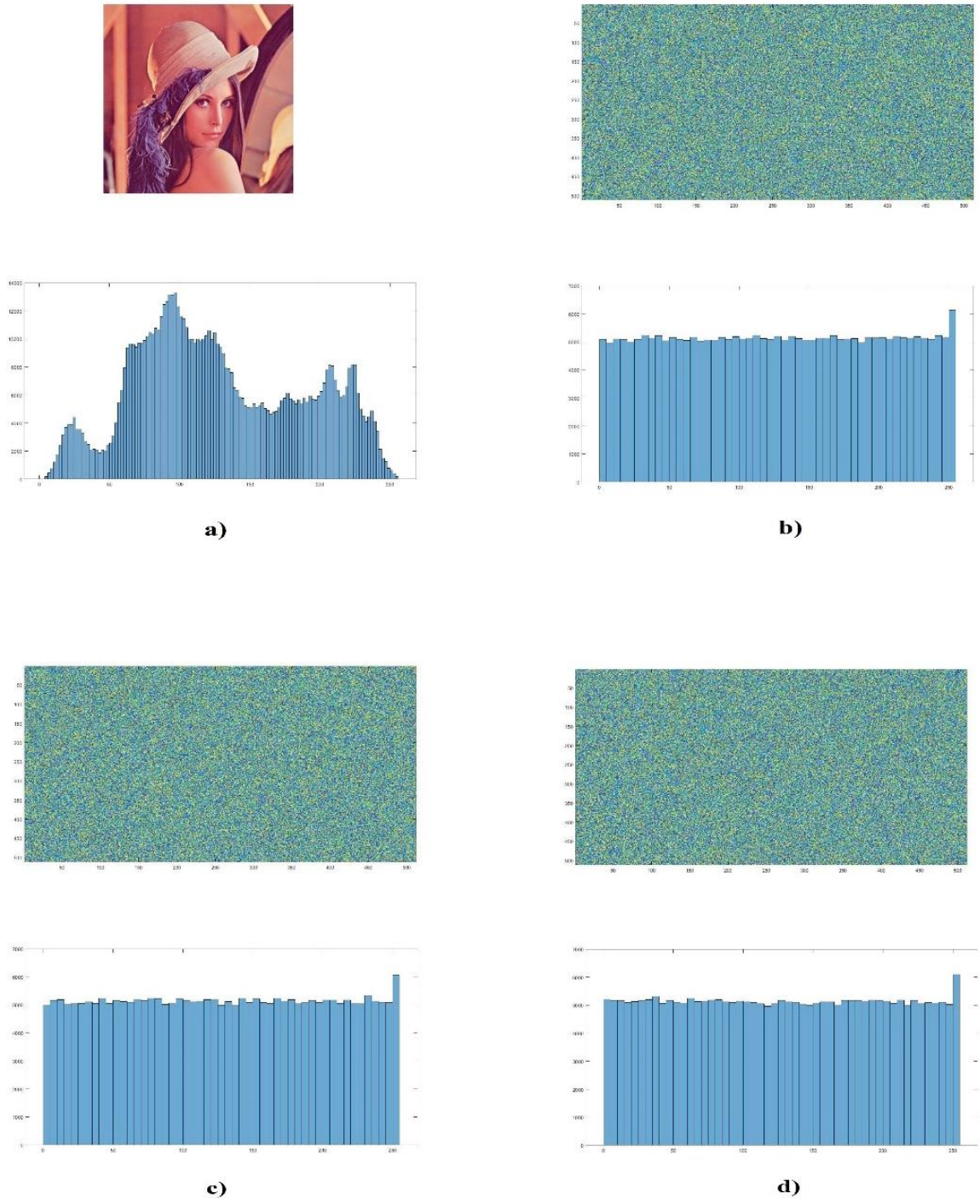
5.7. LCG-Trivium RSÜ ile Şifreleme Uygulamaları

Tasarlanan LCG-Trivium rastgele sayı üretecinin ürettiği şifreleme anahtarları yardımıyla DES, 3DES ve AES algoritmalarıyla görüntü şifreleme işlemleri yapılmıştır. Her bir şifreleme bloğu için kullanılan şifreleme anahtarı uzunlukları Tablo 5.11'de gösterilmiştir.

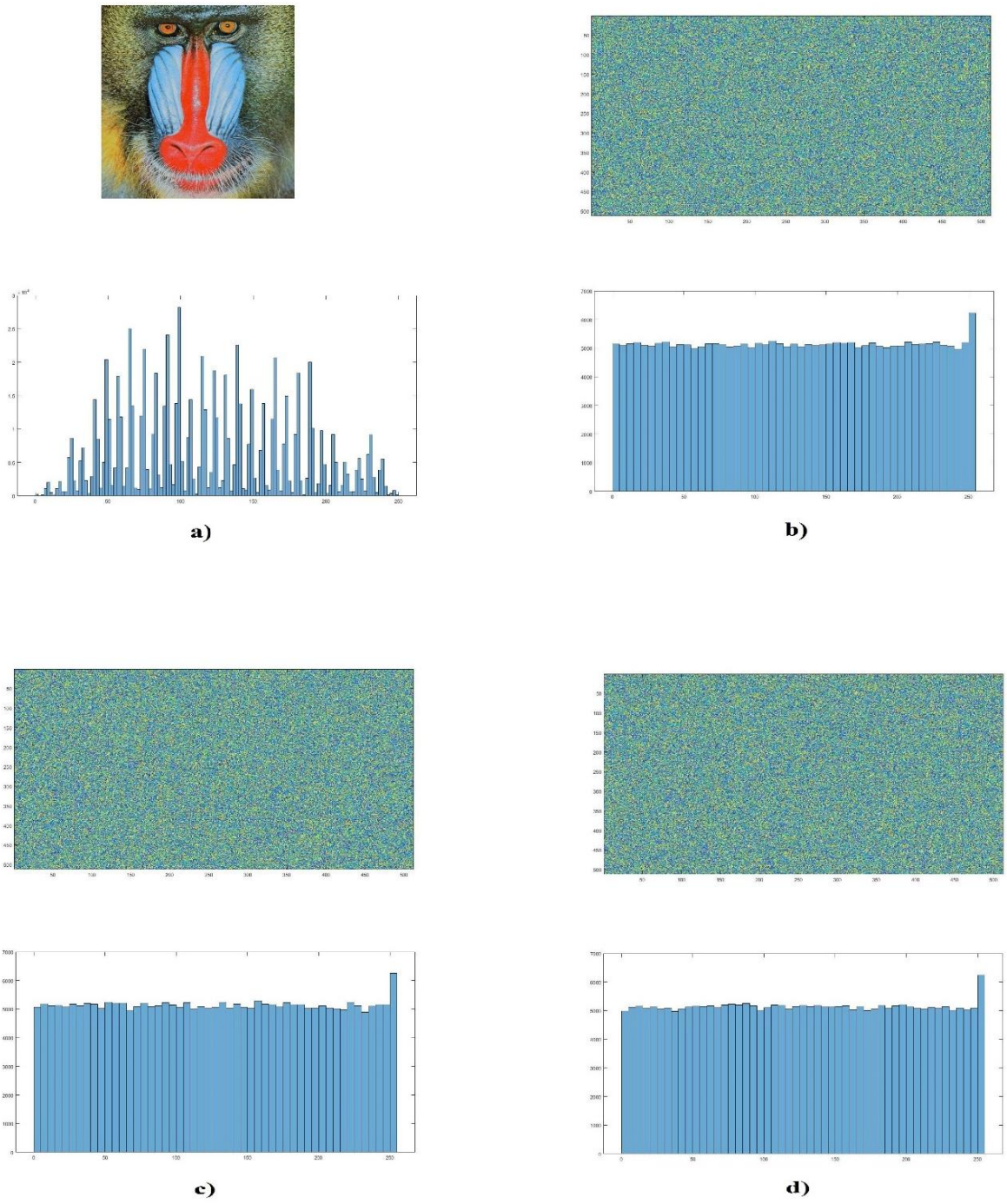
Tablo 5.11. DES, 3DES ve AES anahtar uzunlukları

Şifreleme Algoritması	Anahtar Uzunluğu
DES	64 bit
3DES	192 bit
AES	256 bit

Uygulanan DES, 3DES ve AES şifreleme uygulamaları Şekil 5.17 ve Şekil 5.18'de gösterilmiştir.



Şekil 5.17. LCG-Trivium anahtarlı Lena şifreleme görüntü ve histogram **a)** orijinal görüntü ve histogramı **b)** DES ile şifrelenmiş **c)** 3DES ile şifrelenmiş **d)** AES ile şifrelenmiş



Şekil 5.18. LCG-Trivium anahtarlı Mandrill şifreleme görüntü ve histogramlar **a)** orijinal görüntü **b)** DES ile şifrelenmiş **c)** 3DES ile şifrelenmiş **d)** AES ile şifrelenmiş

Görüldüğü gibi DES, 3DES ve AES başarılı bir şekilde görüntü şifreleme uygulamasında kullanılmıştır. Ancak en temel parametre şifreleme ve şifre çözme süreleridir. Bu üç algoritmaya ilişkin şifreleme ve şifre çözme süreleri Tablo 5.12'de gösterilmiştir.

Tablo 5.12. DES, 3DES ve AES şifreleme ve şifre çözme süreleri

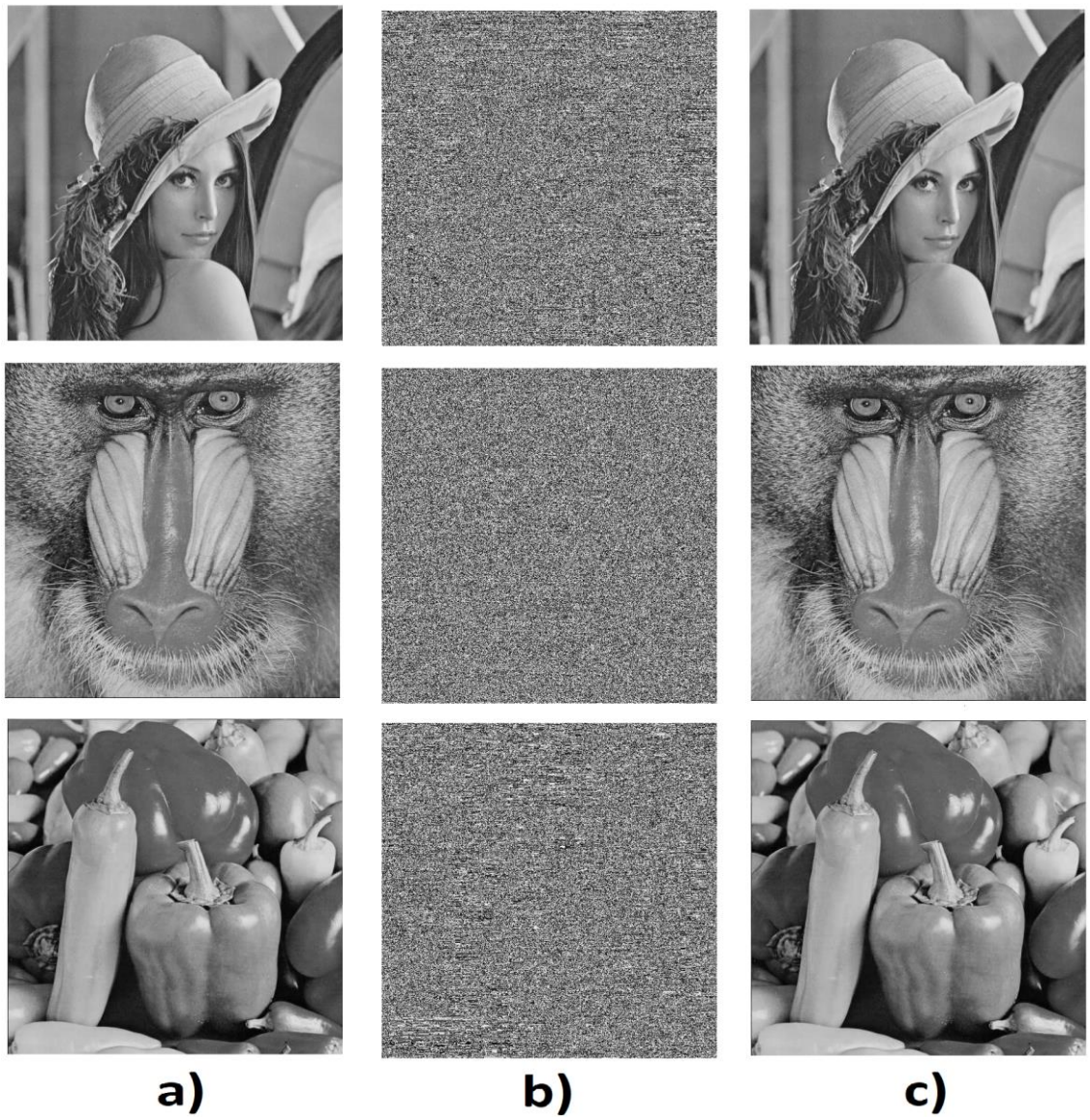
Görüntü Örneği	Seçilen Algoritma	Şifreleme Süresi (ms)
Mandrill 512x512	AES	Şifreleme Süresi: 8,03 ms Şifre Çözme Süresi: 3,27 ms
	DES	Şifreleme Süresi: 10,58 ms Şifre Çözme Süresi: 5,83 ms
	3DES	Şifreleme Süresi: 18,03 ms Şifre Çözme Süresi: 11,94 ms
Lena 512x512	AES	Şifreleme Süresi: 13,71 ms Şifre Çözme Süresi: 6,49 ms
	DES	Şifreleme Süresi: 19,68 ms Şifre Çözme Süresi: 14,11 ms
	3DES	Şifreleme Süresi: 42,05 ms Şifre Çözme Süresi: 36,63 ms

Örnek resimlerle yapılan şifreleme uygulamalarına göre en başarılı şifreleme AES ile yapılmıştır. Çünkü AES son yürürlükteki standart olmasının yanı sıra daha uzun şifreleme anahtarı boyutunu da desteklemektedir. Bu da şifreleme sisteminin güvenliğini oldukça arttıracaktır. Ayrıca şifreleme ve şifre çözme sürelerinin en iyi olduğu algoritma da yine AES'tir. Tüm yönleriyle AES'in başarılı bir şifreleme standardı olduğu söylenebilir.

5.8. Önerilen Simetrik Blok Şifreleme Algoritmasının Uygulanması

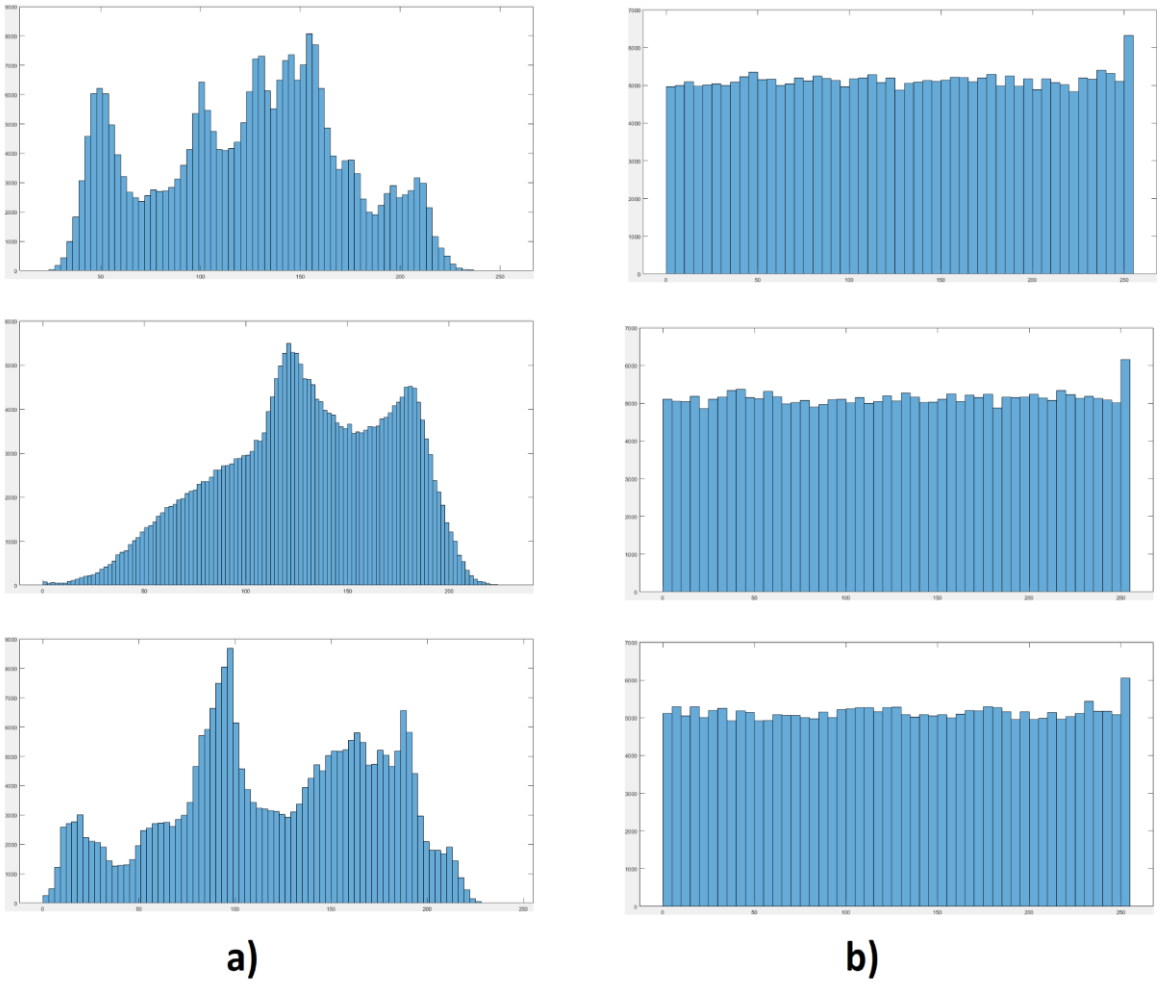
Geliştirmiş olduğumuz simetrik blok şifreleme algoritmasının uygulanması ve güvenlik analizleri bu bölümde incelenmiştir. Geliştirilen şifreleme algoritması diğer simetrik blok şifreleme algoritmaları gibi her türlü veri çeşidine uygun olarak işlem yapabilse de görüntü şifreleme uygulamalarının daha önce bahsedilen avantajlarından dolayı şifreleme işleminde tercih edilmesi daha uygun olacaktır.

Önerilen simetrik blok şifreleme algoritması literatürde görüntü işleme operasyonları için sıklıkla kullanılan Lena, Mandrill ve Peppers örnek görüntülerine uygulanmıştır. Orijinal görüntüler, şifrelenmiş görüntüler ve şifre çözme işlemi uygulanan görüntüler Şekil 5.19'de gösterilmiştir.



Şekil 5.19. Önerilen blok şifreleme algoritmasının uygulanması **a)** orijinal görüntü **b)** şifrelenmiş görüntü **c)** şifresi çözülmüş görüntü

Her üç görüntünün de başarılı bir şekilde şifrelenip, şifre çözme işlemi uygulandığında orijinal görüntüye döndürüldüğü belirlenmiştir. Söz konusu görüntülerin histogram analizi Şekil 5.20'de gösterilmiştir.



Şekil 5.20. Önerilen blok şifreleme algoritması uygulanan görüntülerin histogram analizi **a)** orijinal görüntü **b)** şifrelenmiş görüntü

Histogram analizine göre farklı histogram dağılımlarına sahip görüntüler şifrelendiklerinde sonuç hep düzgün dağılımlı bir histogram olmaktadır. Bu durum şifreleme işlemlerinin oldukça başarılı bir şekilde gerçekleştirildiğini göstermektedir.

Görüntü şifreleme işlemlerine özel bir analiz olan NPCR ve UACI testlerinin sonuçları Tablo 5.13'de gösterilmektedir.

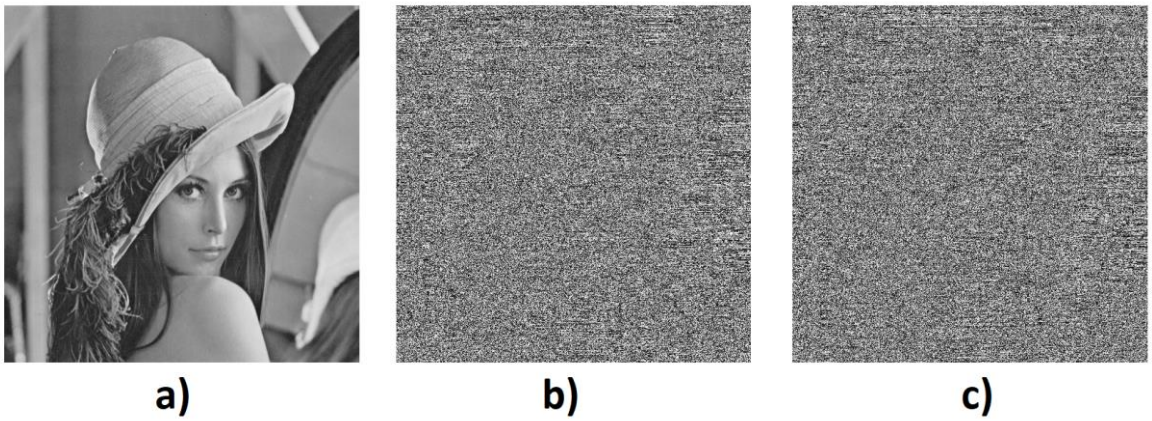
Tablo 5.13. Önerilen şifreleme algoritmasına ilişkin NPCR ve UACI test sonuçlarının karşılaştırılması

Şifreleme Sistemi	**Örnek Görüntü**	**NPCR**	**UACI**
Önerilen Algoritma	Lena	%99,6517	%33,8073
Önerilen Algoritma	Mandrill	%99,1929	%33,8560
Önerilen Algoritma	Peppers	%99,2984	%32,9580
[153]	Lena	%99,84	%34,2
[172]	Lena	%99,6097	%33,4262
[223]	Lena	%99,5693	%33,2824
[224]	Lena	%99,61	%33,53
[224]	Peppers	%99,63	%33,58
[225]	Lena	%99,64	%33,42
[4]	Lena	%99,6015	%33,4555

Karşılaştırma tablosunda görüldüğü gibi NPCR ve UACI testlerinde önerilen şifreleme algoritması başarılı sonuçlar elde etmiştir. Elde edilen test değerleri literatürdeki çalışmalarla da örtüşmektedir.

Şifrelenmiş metin içerisindeki bit oranlarının hesaplanmasında kullanılan T değeri önerilen şifreleme algoritmasının 1000'den fazla örneğe uygulanmasıyla en düşük %49,801 ve en yüksek %50,213 olarak hesaplanmıştır. Sonuçlar %50 civarında ortalamaya sahip olduğu için sistemin başarılı olduğu söylenebilir.

Anahtar hassasiyeti analizinde 512 bit uzunluğundaki simetrik olarak hem şifreleme hem de şifre çözme işlemlerinde kullanılan anahtarın yalnızca bir bit'i değiştirilerek şifre çözme işlemi tekrar uygulandığında Şekil 5.21 elde edilmiştir.



Şekil 5.21. Önerilen şifreleme algoritmasıyla anahtar hassasiyeti analizi

Görüldüğü gibi şifreleme anahtarının sadece bir bit'i bile farklı olduğunda orijinal görüntüyü ya da benzerini elde etmek mümkün olmayacaktır. Bu da önerilen şifreleme algoritması için farklı bir başarı ölçütü olarak öne çıkmaktadır.

Önerilen şifreleme algoritması 512 bit'lik şifreleme anahtarı kullandığı için anahtarın tahmin edilmesi de oldukça güçtür. Literatürde 128 bit ve üstündeki anahtar uzunluklarının belirli bir düzene göre oluşturulmayıp rastgele seçilmesi durumunda tahmin edilmesinin oldukça güç olduğu söylenmektedir [54]. Bu durum bilgisayarların işlem gücünün artmasıyla günden güne geçerliliğini yitirse de 512 bit uzunluğundaki bir anahtarı tahmin etmek 2^{512} bit uzunluğundaki bir sayıyı tahmin etmek gibi olacaktır. Bu da 155 basamaklı bir sayı kadar farklı tahmin olasılığının oluşması demektir.

Gerçekleştirilen şifreleme işlemlerine ilişkin bilgi entropisi (Shannon Entropi) ve üç yönlü korelasyon katsayısı hesaplamaları ile literatür karşılaştırması Tablo 5.14'te gösterildiği şekilde bulunmuştur.

Tablo 5.14. Önerilen şifreleme algoritmasına ilişkin bilgi entropisi ve korelasyon katsayısı değerlerinin karşılaştırılması

Referans	Görüntü	Bilgi Entropisi	Korelasyon Katsayıları		
			Çapraz	Yatay	Dikey
Önerilen	Lena	7,99850	0,00066	0,01156	0,00370
Önerilen	Mandrill	7,99871	0,00180	0,02820	0,00220
Önerilen	Peppers	7,99837	-0,0037	0,01720	-0,0043
[159]	Lena	7,9973	-0,0029	-0,0032	0,0040
[172]	Lena	7,99154	0,00186	-0,0015	0,00185
[223]	Lena	7,9993	0,0189	-0,0097	0,0086
[223]	Peppers	7,9974	-0,0029	-0,0076	0,0039
[223]	Mandrill	7,9993	-0,0151	-0,0153	-0,0057
[224]	Lena	7,9973	-0,0226	0,0041	0,0368
[4]	Lena	7,9974	0,0019	0,0053	0,0042

Görüldüğü gibi elde edilen korelasyon katsayıları literatüre paralel şekilde pozitif ya da negatif yönden sıfıra yakın bir şekilde hesaplanmıştır. Ayrıca bilgi entropisi değerleri ideal değer olan "8"e yakın olarak yine literatürdeki değerlerle uyumlu şekilde hesaplanmıştır.

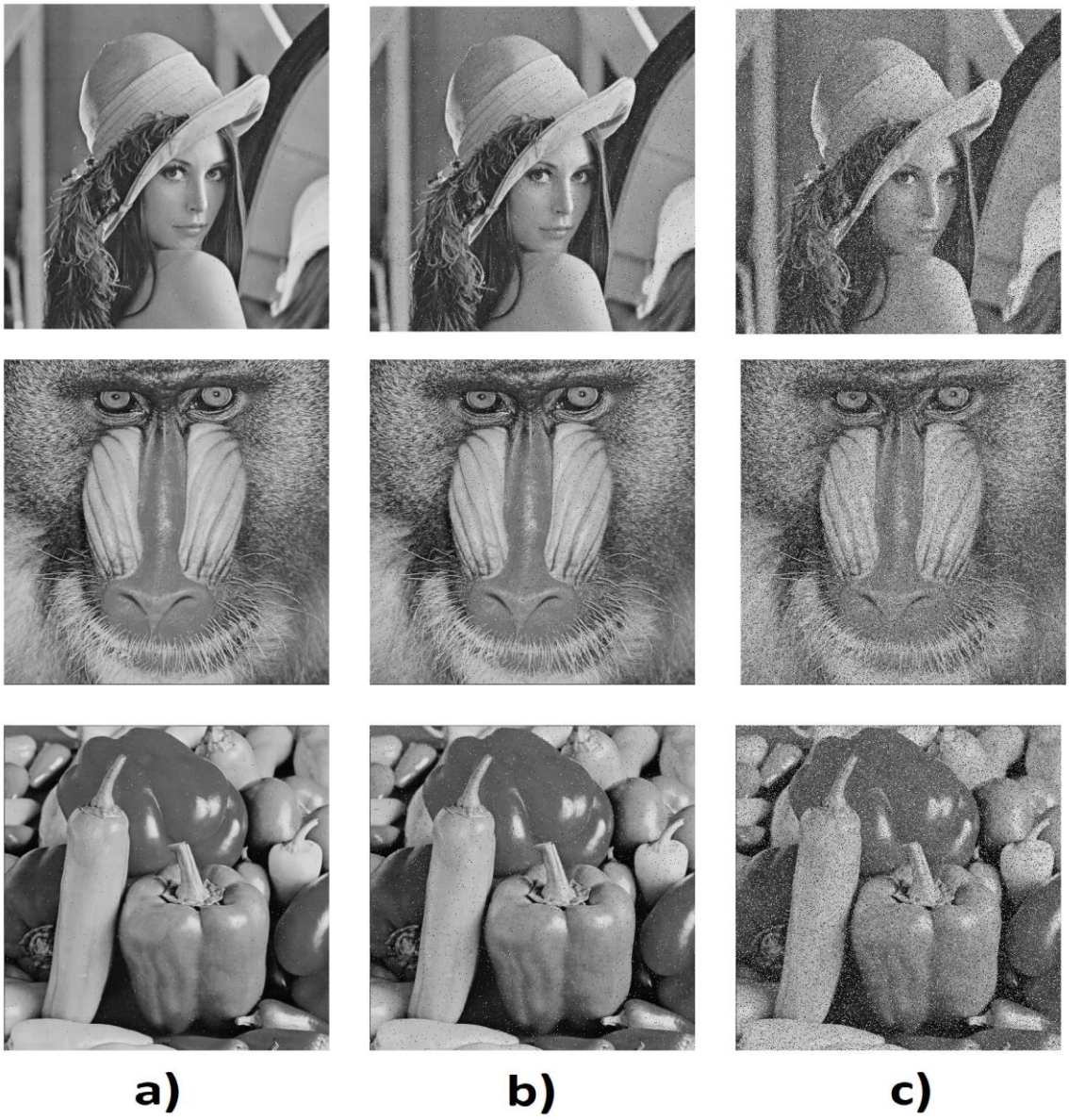
Materyal ve metot bölümünde bahsedilen MSE ve PSNR değerlerinin hesaplanması ve literatür ile karşılaştırılması da Tablo 5.15'de gösterilmiştir.

Tablo 5.15. Önerilen şifreleme algoritmasına ilişkin MSE ve PSNR değerlerinin karşılaştırılması

Şifreleme Sistemi	Örnek Görüntü	MSE Değeri	PSNR Değeri
Önerilen Algoritma	Lena	7760	9,231
Önerilen Algoritma	Mandrill	7234	9,537
Önerilen Algoritma	Peppers	8438	8,868
[158]	Lena	-	9,208
[153]	Lena	11258	7,616
[152]	Mandrill	-	4,485
[226]	Lena	-	9,208
[227]	Lena	7760	9,232
[228]	Lena	10351	9,551

Literatürde bu analize ilişkin değerler paylaşılırken PSNR değerlerine öncelik verilmektedir. Çünkü PSNR değerinin hesaplanması için zaten MSE değeri hesaplanmalıdır. Bu yüzden karşılaştırma tablosunda bazı çalışmaların MSE değerini paylaşmaması normal karşılanabilir. MSE ve PSNR analizine göre de elde edilen değerler literatürdeki çalışmalarla benzerlik göstermektedir. Bu yüzden önerilen simetrik blok şifreleme algoritmasının bu analize göre de başarılı olduğu söylenebilir.

Önerilen sistemin gürültü saldırılarına karşı dayanıklılığını belirlemek için şifrelenmiş metinlere belirli oranlarda gürültü eklenerek şifre çözme işlemleri yapılmıştır. Buna göre farklı yoğunluktaki gürültü değerlerine karşılık şifre çözme işlemi sonucunda elde edilen görüntüler Şekil 5.22'de verilmiştir.



Şekil 5.22. Gürültü saldırısı sonucunda elde edilen görüntüler **a)** d= 0,0005 **b)** d= 0,005 **c)** d= 0,05

Şekil incelendiğinde her bir görüntü grubuna farklı yoğunluktaki gürültüler uygulanarak şifre çözme adımları gerçekleştirilmiştir. Görüntülerde meydana gelen bozulmalar gürültü yoğunluğuna bağlı olarak artsa da orijinal görüntülerin ayırt edilebilmesi mümkündür. En yoğun gürültü olan % 5'lik gürültü de bile görüntülerin ayırt edilebilmesi bu testin başarılı olduğunu göstermektedir.

Önerilen şifreleme sistemi 3.1 GHz işlemci hızına sahip 4 GB Ram içeren bir bilgisayarda lisansı Fırat Üniversitesi tarafından sağlanan Matlab programı yardımıyla gerçekleştirilmiştir. Önerilen sistemin 512x512'lik bir görüntüyü şifrelemesi için gereken süre 0,02001 saniye olarak hesaplanmıştır. Aynı bilgisayarda ve aynı Matlab platformu üzerinde DES, 3DES ve AES algoritmalarına ilişkin şifreleme hızlarının karşılaştırma tablosu Tablo 5.16'da gösterildiği şekilde olmuştur.

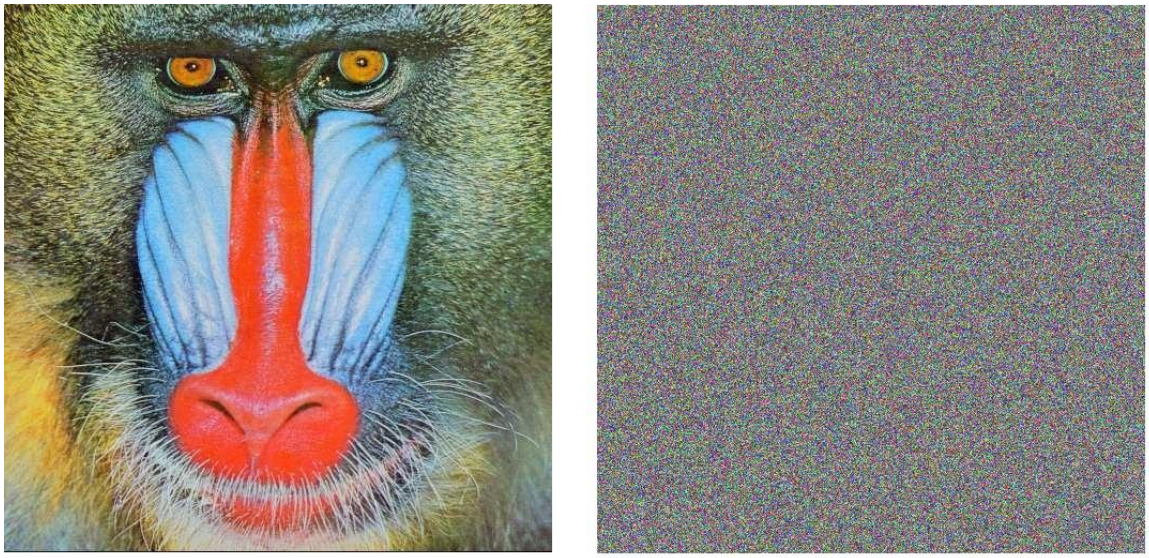
Tablo 5.16. DES, 3DES,AES ile önerilen algoritmanın şifreleme hızları

Şifreleme Algoritması	Ortalama Şifreleme Hızı
DES	34,38 Mbit/s
3DES	13,66 Mbit/s
AES	49,34 Mbit/s
Önerilen Algoritma	105,01 Mbit/s

Tabloda görüldüğü gibi önerilen şifreleme algoritması diğer blok şifreleme algoritmaları baz alındığında çok daha hızlı işlem yapmaktadır. Ayrıca literatürdeki sistemler yıllar içerisindeki birçok araştırmacının katkısıyla en hızlı çalışacak şekilde dizayn edilmiştir. Bilgisayar sistemleri de bu algoritmaların işlem döngülerine uygun olarak üretildiğinden çalışma hızları çok daha yüksektir. Önerilen algoritma ise henüz yaygınlaşmadığı için zaman içerisinde geliştirmelerle çok daha hızlı işlem yapacak hale gelebilir.

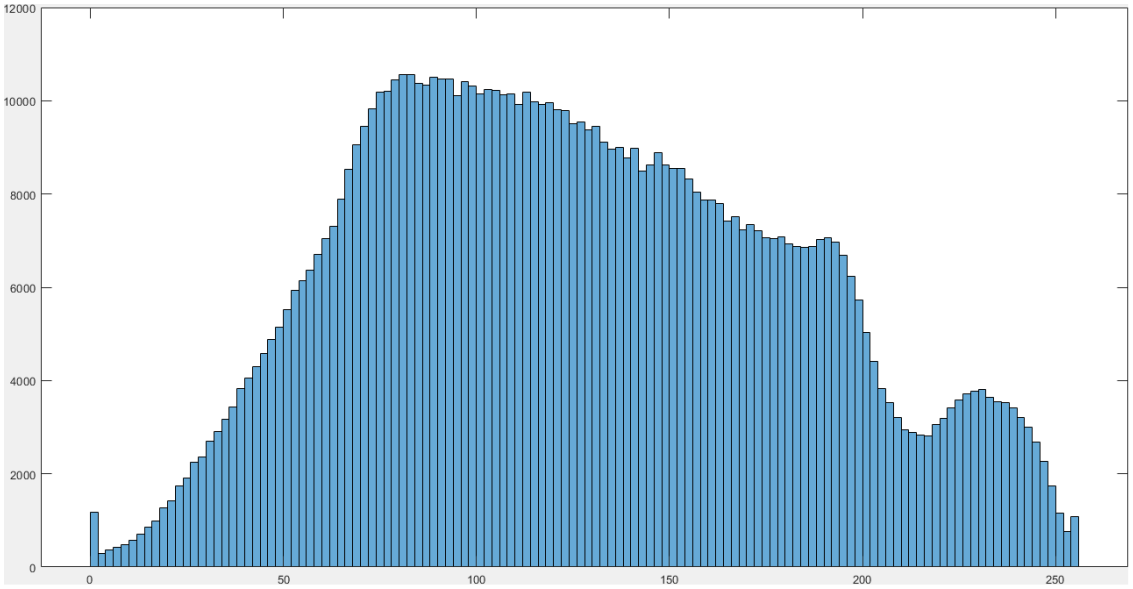
5.9. Önerilen Piksel Karıştırma Uygulaması

Önceki bölümde tasarım detayları gösterilen 512x512 boyutundaki piksel karıştırma matrisi örnek görüntülere uygulanarak piksel karıştırma işlemi yapılmıştır. Örnek bir görünyüe uygulanan piksel karıştırma işlemi Şekil 5.23'te gösterilmiştir.

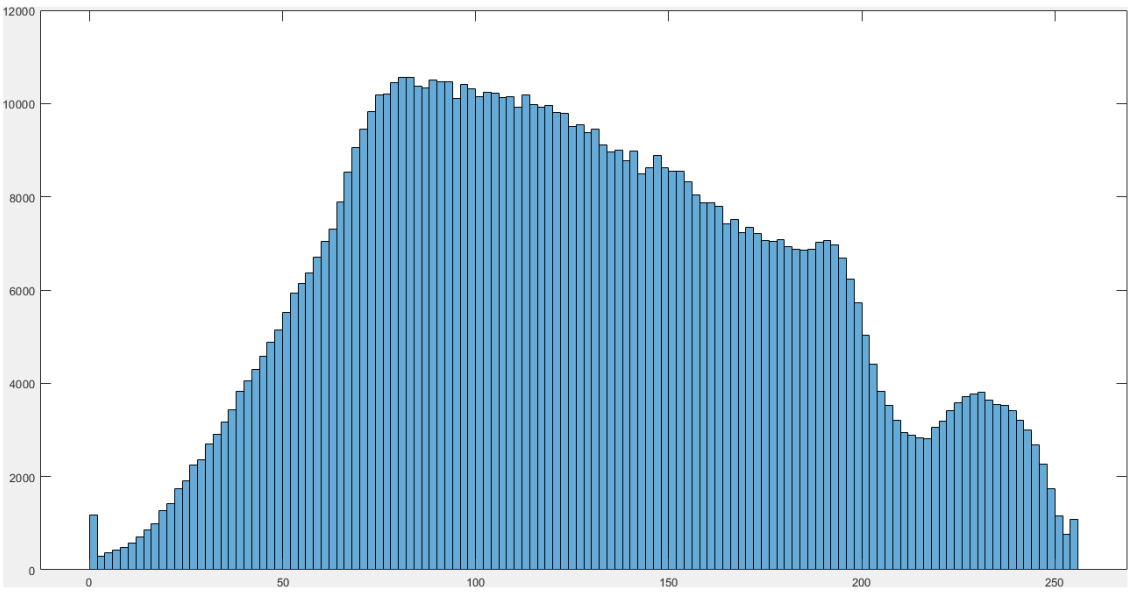


Şekil 5.23. Piksel karıştırma uygulaması **a)** orijinal görüntü **b)** piksel karıştırma uygulanan görüntü

Görüldüğü gibi ilk bakışta piksel karıştırma uygulamasının şifreleme uygulamaları gibi görüntüyü tanınmaz hale getirdiği söylenebilir. Ancak görüntü detaylı olarak incelenirse bu uygulamanın şifreleme sistemlerinin yerini tutamayacağı anlaşılmaktadır. Görüntülere ilişkin histogramlar Şekil 5.24'te gösterilmiştir.



a)

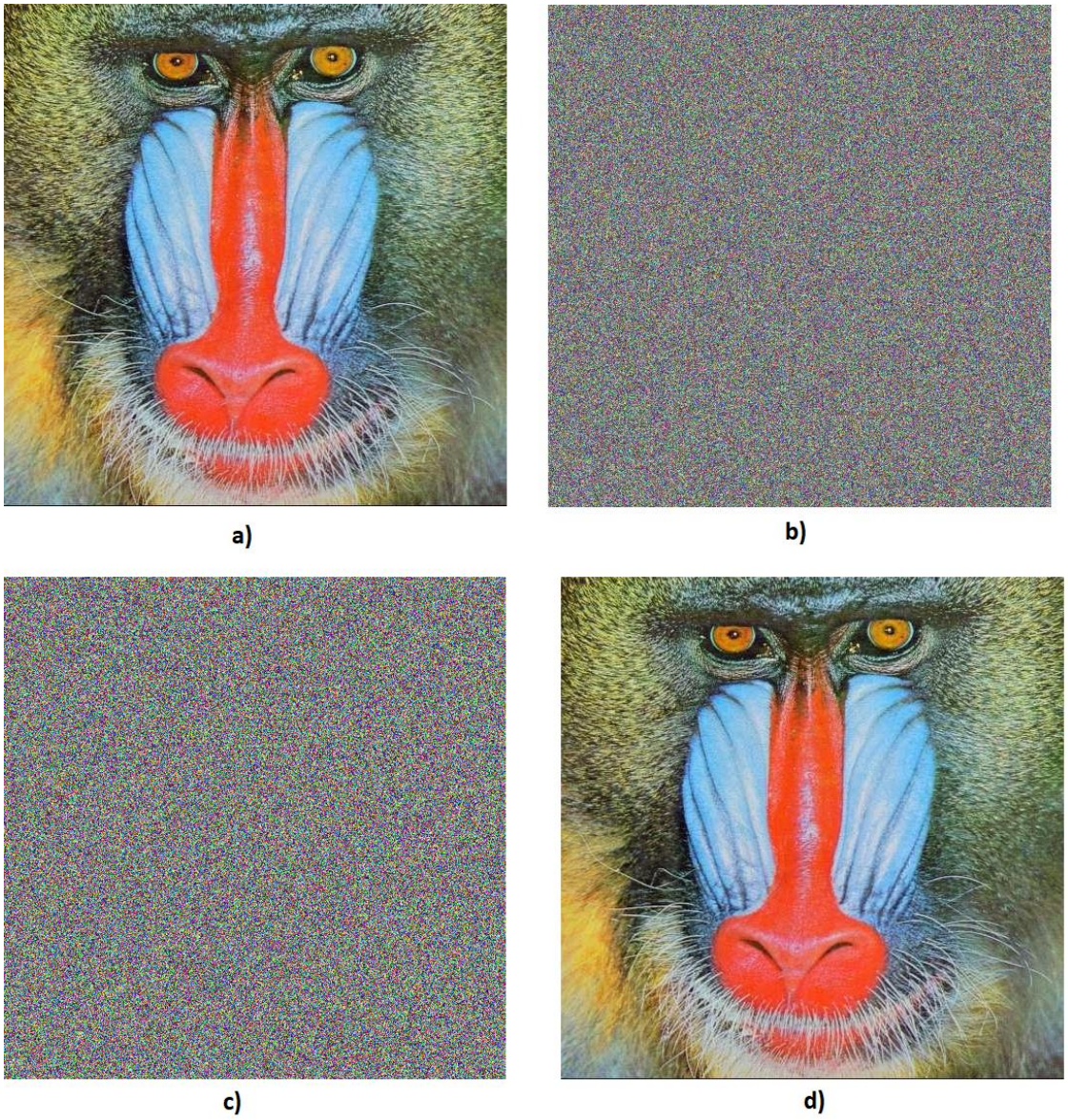


b)

Şekil 5.24. Mandrill görüntüsünün histogram grafikleri **a)** orijinal görüntü **b)** piksel karıştırma uygulanan görüntü

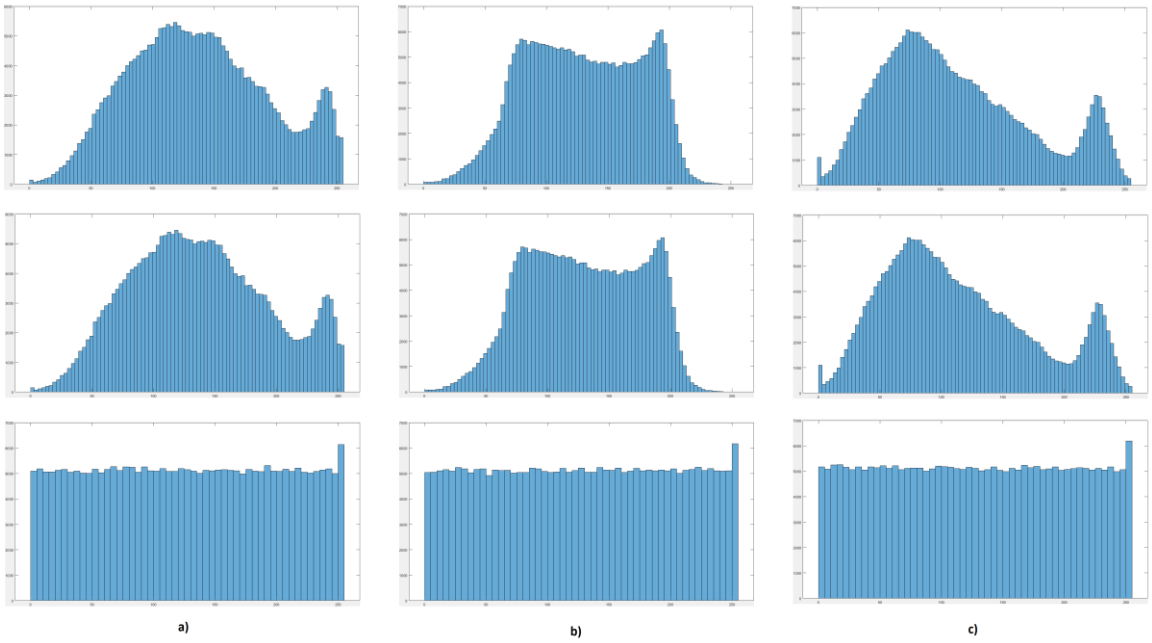
Görüldüğü gibi orijinal görüntü ile piksel karıştırma işlemi uygulanan görüntünün histogram grafikleri birebir aynıdır. Piksel karıştırma işleminde piksellere herhangi bir işlem uygulanmadığı için sadece piksellerin yerleri değiştirildiği için histogram grafikleri aynı kalmıştır.

Piksel karıştırma işlemleri eğer şifreleme sistemlerine ön işlem olarak uygulanırsa sistemin karmaşıklığını arttıracağı için mantıklı olacaktır. Görüntülerde piksel karıştırma işleminden sonra uygulanan önerilen blok şifreleme algoritmasının çıktıları Şekil 5.25'te gösterilmiştir.



Şekil 5.25. Mandrill görüntüsüne piksel karıştırma ve şifreleme işlemlerinin uygulanması **a)** orijinal görüntü, **b)** piksel karıştırma uygulanan görüntü, **c)** şifrelenmiş görüntü, **d)** şifresi çözülmüş ve pikselleri düzeltilmiş görüntü

Başarılı bir şekilde uygulanan şifreleme algoritmasından sonra elde edilen histogram grafikleri Şekil 5.26'da gösterilmiştir.



Şekil 5.26. Renkli görüntüye piksel karıştırma sonrasında şifreleme uygulamasının histogram grafikleri **a)** R katmanı, **b)** G katmanı, **c)** B katmanı

Şekilde ilk sıra orijinal görüntüyü, ikinci sıra piksel karıştırma uygulanan görüntüyü ve üçüncü sıra şifreleme uygulanana görüntünün histogram grafiklerini göstermektedir. Piksel karıştırma işlemi sonrasında histogram grafiklerinde herhangi bir değişiklik olmasa da şifreleme işlemi sonucunda grafikler tekdüze bir yapıya kavuşmuştur.

6. SONUÇLAR

Yapılan tez çalışması neticesinde literatürde ilk kez elektrik alan ölçüm verileriyle bir GRSÜ tasarlanmıştır. Yine literatürde ilk kez kullanılan 2 farklı ses sınıflandırma veri setiyle ve kendi kaydettiğimiz ortam sesleriyle GRSÜ tasarımları yapılmıştır. Doğrusal eşlenik üreteci ve Trivium algoritması birlikte kullanılarak bir SRSÜ tasarımı yapılmıştır. Yine Keccak algoritması ve onun bir alt fonksiyonu olan Farfalle algoritması yardımıyla rastgele bit dizileri üretilerek tüm sistemlerin çeşitli testlerden başarı elde ettiği gösterilmiştir. Üretilen rastgele sayı dizileri için S-kutusu tabanlı bir son işlem algoritmasının uygulaması yapılmıştır. Elde edilen rastgele sayı dizilerini şifreleme anahtarı olarak kullanan Vernam şifreleyicisi, DES, 3DES ve AES algoritmaları yardımıyla ses ve görüntü şifreleme uygulamaları yapılmıştır. Yine rastgele sayı dizilerinden bir rastgele permütasyon üretilmesiyle görüntü piksel karıştırma uygulaması gerçekleştirilmiştir. Tamamen yeni bir blok şifreleme algoritması geliştirilerek farklı kriptanaliz yöntemleriyle başarısı kanıtlanmıştır. Geliştirilen algoritma popüler blok şifreleme algoritmalarıyla karşılaştırılarak önerilen sistemin şifreleme hızı açısından önemli bir avantaj sağladığı gösterilmiştir. Yeni blok şifreleme algoritması içerisinde tamamıyla özgün bir rastgele sayı üreteci tabanlı şifreleme anahtarı üretme sistemi literatüre sunulmuştur. Algoritma içerisinde S-kutuları hem şifreleme sistemi içerisinde kullanılırken hem de şifreleme anahtarı üretilirken kullanılmasının literatür de büyük bir yenilik olacağı düşünülmektedir. Yapılan çalışmaların sonucunda şu çıktılara ulaşılmıştır:

1) Geliştirilen elektrik alan ölçüm değerlerinin GRSÜ tasarımında kullanılmasıyla, rastgelelik şartlarını Lyapunov üstelleri veya Hoshen-Kopelman algoritması gibi testlerle sağlayan her türlü veri seti yardımıyla gerçek rastgele sayı üreteci tasarlamanın mümkün olduğu gösterilmiştir.
2) Yine veri setlerinde bulunan veya herhangi bir ortamdan kaydedilen ses verileriyle başarılı bir gerçek rastgele sayı üreteci tasarımı yapılmıştır.
3) SRSÜ tasarımlarıyla herhangi bir gerçek veri kaynağına ihtiyaç duyulmadan istatistiksel olarak iyi sonuçlar elde eden rastgele sayı üreteçleri uygulanmıştır.
4) Rastgele sayı üreteçleri değerlendirilirken farklı testleri uygulamanın sistemin güvenilirliğini arttıracağı gözlemlenmiştir.
5) Rastgele sayı üreteci çıktılarıyla oluşturulan uygun boyutlu bir permütasyon matrisinin görüntü piksellerinin karıştırılmasında kullanılmıştır.
6) Şifrelenecek veri uzunluğuyla eşit şifreleme anahtarına sahip olan Vernam şifreleyicisi gibi sistemlerin mutlak suretle rastgele sayı dizileri kullanarak şifreleme uygulamalarında kullanılması gerektiği belirlenmiştir. Buna rağmen NPCR ve UACI testi ya da anahtar hassasiyeti analizlerine başarısız kabul edilebilen bu sistemlerde tek kullanımlık rastgele sayı dizisi kullanıldığında güvenli bir şifrelemenin yapılabileceği tespit edilmiştir.

7) DES, 3DES ve AES gibi simetrik blok şifreleme standartlarında rastgele sayı dizileri kullanmanın anahtar tahmin edilebilirliğini düşüreceğinden daha güvenli olduğu belirlenmiştir.
8) Önerilen blok şifreleme sisteminin histogram analizi, NPCR ve UACI testleri, anahtar uzunluğu, anahtar hassasiyeti, bilgi entropisi, korelasyon katsayıları, MSE ve PSNR değerleri gibi kriptanaliz yöntemlerinin tamamına karşı tam başarı sağladığı gösterilmiştir.
9) Önerilen blok şifreleme algoritması içerisindeki şifreleme anahtarı üreteci, özgün yapısıyla ve S-kutularını şifreleme anahtarı üretiminde kullanmasıyla literatüre bir yenilik getirmektedir.
10) Önerilen şifreleme sisteminin iletişim kanalında oluşabilecek gürültülere karşı standart değer olan yüzde 5'e kadar başarılı bir şekilde şifre çözme işlemlerine devam ettiği belirlenmiştir.
11) Önerilen algoritma, şifreleme hızı testlerine göre de standart algoritmalar olan DES, 3DES ve AES'e göre çok daha iyi sonuçlar elde etmiştir.

ÖNERİLER

Yapılan tez çalışması neticesinde farklı uygulamalar gerçekleştirilmiş olsa da bu alanda hala yapılabilecek farklı türde yeni uygulamaların mevcut olduğu görülmektedir. İlerleyen zamanlarda yapılması planlanan çalışmalar şu şekildedir:

1) Mevcut GRSÜ ve SRSÜ yöntemleri dışında literatürde bulunmayan çok daha farklı yeni tasarımların ortaya çıkarılması mümkündür.
2) Literatürde bulunan farklı rastgelelik testleriyle tasarlanan rastgele sayı üreteçlerinin değerlendirilme süreçleri genişletilebilir.
3) Önerilen blok şifreleme algoritmasının kodlanması Matlab platformu dışına çıkartılarak daha etkin bir kodlama ile çok daha hızlı çalışması sağlanabilir.
4) Simetrik blok şifreleme algoritmaları dışında, asimetrik algoritmalar ya da akış şifreleme üzerine yeni algoritmalar tasarlanabilir.
5) Piksel karıştırma işlemleri tamamıyla şifreleme algoritmalarına entegre edilerek farklı şifreleme sistemleri tasarlanabilir.

KAYNAKLAR

[1] A. Akgul, Yeni Kaotik Sistemler ile Rasgele Sayi Üreteci Tasarimi Ve Çoklu-Ortam Verilerinin Yüksek Güvenlikli Şifrelenmesi, Sakarya Universitesi, 2015.

[2] Kamesh, N. Sakthi Priya, A survey of cyber crimes Yanping, Secur. Commun. Networks. 5 (2012) 422–437. https://doi.org/10.1002/sec.

[3] M. Surya Bhupal Rao, V.S. Giridhar Akula, A new chaotic algorithm for image encryption and decryption of digital color images, Int. J. Appl. Eng. Res. 10 (2015) 39217–39222.

[4] H.M. Yuan, Y. Liu, L.H. Gong, J. Wang, A new image cryptosystem based on 2D hyper-chaotic system, Multimed. Tools Appl. 76 (2017) 8087–8108. https://doi.org/10.1007/s11042-016-3454-7.

[5] L. Zhu, H. Song, X. Zhang, M. Yan, T. Zhang, X. Wang, J. Xu, A robust meaningful image encryption scheme based on block compressive sensing and SVD embedding, Signal Processing. 175 (2020) 107629. https://doi.org/10.1016/j.sigpro.2020.107629.

[6] A.A. Alarood, E. Alsolami, M.A. Al-Khasawneh, N. Ababneh, W. Elmedany, IES: Hyper-chaotic plain image encryption scheme using improved shuffled confusion-diffusion, Ain Shams Eng. J. 13 (2022). https://doi.org/10.1016/j.asej.2021.09.010.

[7] S. Çiçek, Y. Uyaroğlu, I. Pehlivan, Simulation and circuit implementation of Sprott Case H chaotic system and its synchronization application for secure communication systems, J. Circuits, Syst. Comput. 22 (2013). https://doi.org/10.1142/S0218126613500229.

[8] L.O. Chua, T. Lin, Chaos in Digital Filters, IEEE Trans. Circuits Syst. 35 (1988) 850–862. https://doi.org/10.1109/31.1832.

[9] I. Pehlivan, Z. Wei, Analysis, nonlinear control, and chaos generator circuit of another strange chaotic system, Turkish J. Electr. Eng. Comput. Sci. 20 (2012) 1229–1239. https://doi.org/10.3906/elk-1103-14.

[10] M. Tuna, C.B. Fidan, A Study on the importance of chaotic oscillators based on FPGA for true random number generating (TRNG) and chaotic systems, J. Fac. Eng. Archit. Gazi Univ. 33 (2018) 413–423. https://doi.org/10.17341/gazimmfd.416355.

[11] B. Karakaya, A. Gülten, M. Frasca, A true random bit generator based on a memristive chaotic circuit: Analysis, design and FPGA implementation, Chaos, Solitons and Fractals. 119 (2019) 143–149. https://doi.org/10.1016/j.chaos.2018.12.021.

[12] B. Karakaya, M.A. Turk, M. Turk, A. Gulten, Selection of optimal numerical method for implementation of Lorenz Chaotic system on FPGA, Int. Adv. Res. Eng. J. 02 (2018) 147–152. www.dergipark.gov.tr.

[13] M. Alçın, M. Tuna, P. Erdoğmuş, İ. Koyuncu, FPGA-based Dual Core TRNG Design Using Ring and Runge-Kutta-Butcher based on Chaotic Oscillator, Chaos Theory Appl. (2021) 20–28. https://doi.org/10.51537/chaos.783548.

[14] L. Merah, A. Ali-pacha, N.H. Said, M. Mamat, Design and FGPA implementation of Lorenz chaotic system for information security issues, Appl. Math. Sci. 7 (2013) 237–246. https://doi.org/10.12988/ams.2013.13022.

[15] B. Bao, L. Xu, N. Wang, H. Bao, Q. Xu, M. Chen, Third-order RLCM-four-elements-based chaotic circuit and its coexisting bubbles, AEU - Int. J. Electron. Commun. 94 (2018) 26–35. https://doi.org/10.1016/j.aeue.2018.06.042.

[16] H. Guler, V. Celik, T. Kaya, Y. Erol, The real time implementation of a chaotic system's synchronization for secure communication, Teh. Vjesn. 25 (2018) 43–48. https://doi.org/10.17559/TV-20160420113930.

[17] E. Avaroğlu, The implementation of ring oscillator based PUF designs in Field Programmable Gate Arrays using of different challenge, Phys. A Stat. Mech. Its Appl. 546 (2020) 124291. https://doi.org/10.1016/j.physa.2020.124291.

[18] S. Yakut, T. Tuncer, A.B. Özer, A New Secure and Efficient Approach for TRNG and Its Post-Processing Algorithms, J. Circuits, Syst. Comput. 29 (2020). https://doi.org/10.1142/S0218126620502448.

[19] M.S. Kim, I.W. Tcho, S.J. Park, Y.K. Choi, Random number generator with a chaotic wind-driven triboelectric energy harvester, Nano Energy. 78 (2020) 105275. https://doi.org/10.1016/j.nanoen.2020.105275.

[20] S. Saravanan, M. Sivabalakrishnan, A hybrid chaotic map with coefficient improved whale optimization-based parameter tuning for enhanced image encryption, Soft Comput. 25 (2021) 5299–5322. https://doi.org/10.1007/s00500-020-05528-w.

[21] B. Gong, H. Zhang, L. Wan, Generating Chaotic Series via Encryption Method in Fractional-Order Chua Systems, Discret. Dyn. Nat. Soc. 2021 (2021). https://doi.org/10.1155/2021/6653930.

[22] T. Tuncer, Implementation of duplicate TRNG on FPGA by using two different randomness source, Elektron. Ir Elektrotechnika. 21 (2015) 35–39. https://doi.org/10.5755/j01.eee.21.4.12779.

[23] E. Avaroglu, Pseudorandom number generator based on Arnold cat map and statistical analysis, Turkish J. Electr. Eng. Comput. Sci. 25 (2017) 633–643. https://doi.org/10.3906/elk-1507-253.

[24] H. Jiang, D. Belkin, S.E. Savel'Ev, S. Lin, Z. Wang, Y. Li, S. Joshi, R. Midya, C. Li, M. Rao, M. Barnell, Q. Wu, J.J. Yang, Q. Xia, A novel true random number generator based on a stochastic diffusive memristor, Nat. Commun. 8 (2017). https://doi.org/10.1038/s41467-017-00869-x.

[25] T. Kaya, Memristor and Trivium-based true random number generator, Phys. A Stat. Mech. Its Appl. 542 (2020) 124071. https://doi.org/10.1016/j.physa.2019.124071.

[26] F. Pareschi, G. Setti, R. Rovatti, Implementation and testing of high-speed CMOS true random number generators based on chaotic systems, IEEE Trans. Circuits Syst. I Regul. Pap. 57 (2010) 3124–3137. https://doi.org/10.1109/TCSI.2010.2052515.

[27] A. Akgul, S. Kaçar, I. Pehlivan, An Audio Data Encryption with Single and Double Dimension Discrete-Time Chaotic Systems, Tojsat. 5 (2015) 14–23.

[28] L. Akçay, E. Cil, A. Vardar, I. Yaman, R. Yeniceri, M.E. Yalcin, Implementation of a chaotic time-delay RNG based secure communication system on FPGA, in: 2017 10th Int. Conf. Electr. Electron. Eng. ELECO 2017, 2018: pp. 1277–1280.

[29] J. Li, Z.L. Wang, H. Zhao, R. Gravina, G. Fortino, Y. Jiang, K. Tang, Networked human motion capture system based on quaternion navigation, 2017. https://doi.org/10.1145/0000000.0000000.

[30] S. Kadhe, B. Garcia, A. Heidarzadeh, S. El Rouayheb, A. Sprintson, Private Information Retrieval

with Side Information, 2020. https://doi.org/10.1109/TIT.2019.2948845.

[31] B. Karakaya, V. Çelik, A. Gülten, Chaotic cellular neural network-based true random number generator, Int. J. Circuit Theory Appl. 45 (2017) 1885–1897. https://doi.org/10.1002/cta.2374.

[32] S.A. Tuncer, Real-time random number generation with ring oscillator based double physically unclonable function, Inf. MIDEM. 48 (2018) 121–128.

[33] M. Babaei, M. Farhadi, Introduction to Secure PRNGs, Int. J. Commun. Netw. Syst. Sci. 04 (2011) 616–621. https://doi.org/10.4236/ijcns.2011.410074.

[34] U. Guler, A.E. Pusane, G. Dundar, Design of efficient CMOS ring oscillator-based random number generator, Int. J. Electron. 104 (2017) 1465–1482. https://doi.org/10.1080/00207217.2017.1312704.

[35] S. Buchovecká, R. Lórencz, F. Kodýtek, J. Buček, True random number generator based on ring oscillator PUF circuit, Microprocess. Microsyst. 53 (2017) 33–41. https://doi.org/10.1016/j.micpro.2017.06.021.

[36] K. Bhattacharjee, D. Paul, S. Das, Pseudo-random number generation using a 3-state cellular automaton, Int. J. Mod. Phys. C. 28 (2017). https://doi.org/10.1142/S0129183117500784.

[37] D. Nguyen, D. Tran, W. Ma, D. Sharma, Random number generators based on EEG non-linear and chaotic characteristics, J. Cyber Secur. Mobil. 6 (2017) 305–338. https://doi.org/10.13052/jcsm2245-1439.634.

[38] S.R. Moosavi, E. Nigussie, S. Virtanen, J. Isoaho, Cryptographic key generation using ECG signal, 2017 14th IEEE Annu. Consum. Commun. Netw. Conf. CCNC 2017. (2017) 1024–1031. https://doi.org/10.1109/CCNC.2017.7983280.

[39] S. Arslan Tuncer, T. Kaya, True Random Number Generation from Bioelectrical and Physical Signals, Comput. Math. Methods Med. 2018 (2018) 1–11. https://doi.org/10.1155/2018/3579275.

[40] B. Petchlert, H. Hasegawa, Using a low-cost electroencephalogram (EEG) directly as random number generator, Proc. - 2014 IIAI 3rd Int. Conf. Adv. Appl. Informatics, IIAI-AAI 2014. (2014) 470–474. https://doi.org/10.1109/IIAI-AAI.2014.100.

[41] G. Chen, Are electroencephalogram (EEG) signals pseudo-random number generators?, J. Comput. Appl. Math. 268 (2014) 1–4. https://doi.org/10.1016/j.cam.2014.02.028.

[42] M.M. Abutaleb, A novel true random number generator based on QCA nanocomputing, Nano Commun. Netw. 17 (2018) 14–20. https://doi.org/10.1016/j.nancom.2018.04.001.

[43] S. Ramakrishnan, Cryptographic and Information Security, 2018. https://doi.org/10.1201/9780429435461.

[44] V.K. Rai, S. Tripathy, J. Mathew, Memristor based Random Number Generator: Architectures and Evaluation, Procedia Comput. Sci. 125 (2018) 576–583. https://doi.org/10.1016/j.procs.2017.12.074.

[45] H.S. Alhadawi, M.F. Zolkipli, S.M. Ismail, D. Lambić, Designing a pseudorandom bit generator based on LFSRs and a discrete chaotic map, Cryptologia. 43 (2019) 190–211. https://doi.org/10.1080/01611194.2018.1548390.

[46] S. Coşkun, İ. Pehlivan, A. Akgül, B. Gürevin, A new computer-controlled platform for ADC-based true random number generator and its applications, Turkish J. Electr. Eng. Comput. Sci. 27 (2019) 847–860. https://doi.org/10.3906/elk-1806-167.

[47] C. Guo, Y. Zhou, A True Random Number Generator Based on the Multiple-Scrolls Chaotic System,

Proc. - 2018 3rd Int. Conf. Inf. Syst. Eng. ICISE 2018. (2019) 98–103. https://doi.org/10.1109/ICISE.2018.00026.

[48] R.A. Elmanfaloty, E. Abou-Bakr, Random property enhancement of a 1D chaotic PRNG with finite precision implementation, Chaos, Solitons and Fractals. 118 (2019) 134–144. https://doi.org/10.1016/j.chaos.2018.11.019.

[49] G. Di Patrizio Stanchieri, A. De Marcellis, E. Palange, M. Faccio, A true random number generator architecture based on a reduced number of FPGA primitives, AEU - Int. J. Electron. Commun. 105 (2019) 15–23. https://doi.org/10.1016/j.aeue.2019.03.006.

[50] T. Etem, T. Kaya, Trivium-Linear Congruential Generator Based Bit Generation For Image Encryption, Fırat Üniversitesi Mühendislik Bilim. Derg. 32 (2020) 287–294. https://doi.org/10.35234/fumbd.687403.

[51] J.T. Kavulich, B.P. Van Deren, M. Schlosshauer, Searching for evidence of algorithmic randomness and incomputability in the output of quantum random number generators, Phys. Lett. Sect. A Gen. At. Solid State Phys. 388 (2021) 127032. https://doi.org/10.1016/j.physleta.2020.127032.

[52] S. Cang, Z. Kang, Z. Wang, Pseudo-random number generator based on a generalized conservative Sprott-A system, Nonlinear Dyn. (2021). https://doi.org/10.1007/s11071-021-06310-9.

[53] F. Neugebauer, I. Polian, J.P. Hayes, S-box-based random number generation for stochastic computing, Microprocess. Microsyst. 61 (2018) 316–326. https://doi.org/10.1016/j.micpro.2018.06.009.

[54] X. Chen, S. Qian, F. Yu, Z. Zhang, H. Shen, Y. Huang, S. Cai, Z. Deng, Y. Li, S. Du, Pseudorandom Number Generator Based on Three Kinds of Four-Wing Memristive Hyperchaotic System and Its Application in Image Encryption, Complexity. 2020 (2020). https://doi.org/10.1155/2020/8274685.

[55] J. Li, J. Zheng, P. Whitlock, Efficient deterministic andnon- deterministic pseudorandomnumber generation, Math. Comput. Simul. 143 (2018) 114–124.

[56] K. Bhattacharjee, K. Maity, S. Das, A Search for Good Pseudo-random Number Generators : Survey and Empirical Studies, (2018). http://arxiv.org/abs/1811.04035.

[57] L. Pasqualini, M. Parton, Pseudo Random Number Generation: A Reinforcement Learning approach, Procedia Comput. Sci. 170 (2020) 1122–1127. https://doi.org/10.1016/j.procs.2020.03.057.

[58] F. Özkaynak, Cryptographically secure random number generator with chaotic additional input, Nonlinear Dyn. 78 (2014) 2015–2020. https://doi.org/10.1007/s11071-014-1591-y.

[59] H. Ogras, M. Turk, Digital Image Encryption Scheme using Chaotic Sequences with a Nonlinear Function, in: World Acad. Sci. Eng. Technol., Stockholm, 2012: pp. 885–888.

[60] H. Ogras, M. Turk, S. Ogras, Kaos Tabanlı Sayısal Csk ve Dcsk Modülasyon Tekniklerinin Matlab/Simulink Ortamında Gerçekleştirilmesi, in: IV. İletişim Teknol. Ulus. Sempozyumu, Adana, 2009.

[61] A. Maysaa, Q. Iman, A Speech Encryption Using Chaotic Map and Blowfish Algorithms, J. Basrah Res. 39 (2013) 68–76.

[62] X. Zhang, L. Min, A generalized chaos synchronization based encryption algorithm for sound signal communication, Circuits, Syst. Signal Process. 24 (2005) 535–548. https://doi.org/10.1007/s00034-005-2405-8.

[63] V.R. Falmari, M. Brindha, Privacy preserving biometric authentication using Chaos on remote untrusted server, Meas. J. Int. Meas. Confed. 177 (2021) 109257. https://doi.org/10.1016/j.measurement.2021.109257.

[64] R. Gnanajeyaraman, K. Prasadh, Audio encryption using higher dimensional chaotic map, Int. J. Recent Trends Eng. 1 (2009) 103–107.

[65] S. Fahmy, Encryption and Decryption of Audio Signal based on RSA Algorithm, Int. J. Eng. Technol. Manag. Res. 5 (2018) 57–64. https://doi.org/10.5281/zenodo.1341956.

[66] T. Etem, T. Kaya, A novel True Random Bit Generator design for image encryption, Phys. A Stat. Mech. Its Appl. 540 (2020). https://doi.org/10.1016/j.physa.2019.122750.

[67] F. Özkaynak, Brief review on application of nonlinear dynamics in image encryption, Nonlinear Dyn. 92 (2018) 305–313. https://doi.org/10.1007/s11071-018-4056-x.

[68] M. Garcia-Bosque, A. Pérez, C. Sánchez-Azqueta, S. Celma, Application of a MEMS-based TRNG in a chaotic stream cipher, Sensors (Switzerland). 17 (2017). https://doi.org/10.3390/s17030646.

[69] P. Sathiyamurthi, S. Ramakrishnan, Speech encryption using chaotic shift keying for secured speech communication, Eurasip J. Audio, Speech, Music Process. 2017 (2017). https://doi.org/10.1186/s13636-017-0118-0.

[70] M. Kumari, S. Gupta, P. Sardana, A Survey of Image Encryption Algorithms, 3D Res. 8 (2017). https://doi.org/10.1007/s13319-017-0148-5.

[71] G. Deco, E.T. Rolls, Sequential memory: A putative neural and synaptic dynamical mechanism, 2005. https://doi.org/10.1162/0898929053124875.

[72] N.S. Atalay, Ş. Doğan, T. Tuncer, E. Akbal, İmge Şifreleme Yöntem ve Algoritmaları, Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Derg. 10 (2019) 815–831. https://doi.org/10.24012/DUMF.478877.

[73] N. Gao, X. Jing, S. Lv, J. Mu, L. Sun, Wireless physical layer characteristics based random number generator: Hijack attackers, IEEE Veh. Technol. Conf. 2017-Septe (2018) 1–5. https://doi.org/10.1109/VTCFall.2017.8288225.

[74] F.E. Yardim, E. Afacan, Lorenz-tabanli diferansiyel kaos kaydirmali anahtarlama (DCSK) modeli kullanilarak kaotik bir haberleşme sisteminin simülasyonu, J. Fac. Eng. Archit. Gazi Univ. 25 (2010) 101–110.

[75] M.I. Sobhy, A.R. Shehata, Chaotic algorithms for data encryption, in: ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc., 2001: pp. 997–1000. https://doi.org/10.1109/icassp.2001.941085.

[76] A. Akgul, H. Calgan, I. Koyuncu, I. Pehlivan, A. Istanbullu, Chaos-based engineering applications with a 3D chaotic system without equilibrium points, Nonlinear Dyn. 84 (2016) 481–495. https://doi.org/10.1007/s11071-015-2501-7.

[77] A. Prabu, S. Apprao, M. Jaganmohan, R. Babu, Audio Encryption in Handsets, Int. J. Comput. Appl. 40 (2012) 40–45.

[78] M.E. Sahin, Z.G. Cam, H. Guler, S.E. Hamamci, Simulation and Circuit Implementation Memristive Chaotic System and Its Application for Secure Communication Systems, Sensors Actuators A Phys. 290 (2019) 107–118. https://doi.org/10.1016/j.sna.2019.01.008.

[79] M.I. Khalil, Real-Time Encryption/Decryption of Audio Signal, Int. J. Comput. Netw. Inf. Secur. 8 (2016) 25–31. https://doi.org/10.5815/ijcnis.2016.02.03.

[80] X. Shu, J. Wang, H. Wang, X. Yang, Chaotic direct sequence spread spectrum for secure underwater acoustic communication, Appl. Acoust. 104 (2016) 57–66. https://doi.org/10.1016/j.apacoust.2015.10.015.

[81] M. Khan, F. Masood, A. Alghafis, Secure image encryption scheme based on fractals key with Fibonacci series and discrete dynamical system, Neural Comput. Appl. 32 (2020) 11837–11857. https://doi.org/10.1007/s00521-019-04667-y.

[82] J. Sun, C. Li, T. Lu, A. Akgul, F. Min, A Memristive Chaotic System with Hypermultistability and Its Application in Image Encryption, IEEE Access. 8 (2020) 139289–139298. https://doi.org/10.1109/ACCESS.2020.3012455.

[83] P. Fang, H. Liu, C. Wu, M. Liu, A Secure Chaotic Block Image Encryption Algorithm Using Generative Adversarial Networks and DNA Sequence Coding, Math. Probl. Eng. 2021 (2021). https://doi.org/10.1155/2021/6691547.

[84] İ. Koyuncu, A. Turan Özcerit, The design and realization of a new high speed FPGA-based chaotic true random number generator, Comput. Electr. Eng. 58 (2017) 203–214. https://doi.org/10.1016/j.compeleceng.2016.07.005.

[85] D. Schellekens, B. Preneel, I. Verbauwhede, FPGA vendor agnostic true random number generator, in: Proc. - 2006 Int. Conf. F. Program. Log. Appl. FPL, Madrid, 2006: pp. 139–144. https://doi.org/10.1109/FPL.2006.311206.

[86] M. Dichtl, J. Golic, High-speed TRNG with logic gates only, Lect. Notes Comput. Sci. 4727 (2007) 45–62.

[87] K. Wold, C.H. Tan, Analysis and enhancement of random number generator in FPGA based on oscillator rings, Proc. - 2008 Int. Conf. Reconfigurable Comput. FPGAs, ReConFig 2008. 2009 (2008) 385–390. https://doi.org/10.1109/ReConFig.2008.17.

[88] P.Z. Wieczorek, K. Golofit, Dual-metastability time-competitive true random number generator, IEEE Trans. Circuits Syst. I Regul. Pap. 61 (2014) 134–145. https://doi.org/10.1109/TCSI.2013.2265952.

[89] V. Fischer, M. Drutarovský, M. Šimka, N. Bochard, High performance True Random Number Generator in altera stratix FPLDs, Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 3203 (2004) 555–564. https://doi.org/10.1007/978-3-540-30117-2_57.

[90] H. István, A. Suciu, O. Creţ, FPGA based TRNG using automatic calibration, Proc. - 2009 IEEE 5th Int. Conf. Intell. Comput. Commun. Process. ICCP 2009. (2009) 373–376. https://doi.org/10.1109/ICCP.2009.5284733.

[91] I. Cicek, A.E. Pusane, G. Dundar, A novel design method for discrete time chaos based true random number generators, Integr. VLSI J. 47 (2014) 38–47. https://doi.org/10.1016/j.vlsi.2013.06.003.

[92] J.L. Danger, S. Guilley, P. Hoogvorst, High speed true random number generator based on open loop structures in FPGAs, Microelectronics J. 40 (2009) 1650–1656. https://doi.org/10.1016/j.mejo.2009.02.004.

[93] T. Tuncer, E. Avaroğlu, M. Türk, A.B. Ozer, Implementation of non-periodic sampling true random number generator on fpga, Inf. MIDEM. 44 (2014) 296–302.

[94] I. Cicek, A.E. Pusane, G. Dundar, A new dual entropy core true random number generator, Analog Integr. Circuits Signal Process. 81 (2014) 61–70. https://doi.org/10.1007/s10470-014-0324-y.

[95] P. Kohlbrenner, K. Gaj, An embedded true random number generator for FPGAs, in: ACM/SIGDA Int. Symp. F. Program. Gate Arrays - FPGA, New York, 2004: pp. 71–78. https://doi.org/10.1145/968280.968292.

[96] D.R. Stinson, A Provably Secure True Random Number Generator with Built-in Tolerance to Active Attacks, IEEE Trans. Comput. 56 (2007) 109–119. https://doi.org/10.1109/TC.2007.250627.

[97] E. Avaroğlu, T. Tuncer, A.B. Özer, M. Türk, A new method for hybrid pseudo random number generator, Inf. MIDEM. 44 (2014) 303–311.

[98] L. Merah, A. Ali-Pacha, N.H. Said, M. Mamat, A pseudo random number generator based on the chaotic system of chua's circuit, and its real time FPGA implementation, Appl. Math. Sci. 7 (2013) 2719–2734. https://doi.org/10.12988/ams.2013.13242.

[99] E. Avaroğlu, T. Tuncer, A.B. Özer, B. Ergen, M. Türk, A novel chaos-based post-processing for TRNG, Nonlinear Dyn. 81 (2015) 189–199. https://doi.org/10.1007/s11071-015-1981-9.

[100] M.A. Zidan, A.G. Radwan, K.N. Salama, Random number generation based on digital differential chaos, in: Midwest Symp. Circuits Syst., Seoul, 2011: pp. 1–4. https://doi.org/10.1109/MWSCAS.2011.6026266.

[101] M.S. Salah Azzaz, C. Tanougast, S. Sadoudi, R. Fellah, A. Dandache, A new auto-switched chaotic system and its FPGA implementation, Commun. Nonlinear Sci. Numer. Simul. 18 (2013) 1792–1804. https://doi.org/10.1016/j.cnsns.2012.11.025.

[102] İ. Koyuncu, Kriptolojik uygulamalar için FPGA tabanlı yeni kaotik osilatörlerin ve gerçek rasgele sayı üreteçlerinin tasarımı ve gerçeklenmesi, Sakarya Üniversitesi, 2014. https://tez.yok.gov.tr/UlusalTezMerkezi/tezSorguSonucYeni.jsp.

[103] M. Alçin, I. Pehlivan, I. Koyuncu, Hardware design and implementation of a novel ANN-based chaotic generator in FPGA, Optik (Stuttg). 127 (2016) 5500–5505. https://doi.org/10.1016/j.ijleo.2016.03.042.

[104] M. Tuna, C.B. Fidan, I. Koyuncu, I. Pehlivan, Real time hardware implementation of the 3D chaotic oscillator which having golden-section equilibra, in: IEEE Sinyal İşleme ve İletişim Uygulamaları Kurult., 2016: pp. 1309–1312. https://doi.org/10.1109/siu.2016.7495988.

[105] J.M. Bahi, X. Fang, C. Guyeux, An optimization technique on pseudorandom generators based on chaotic iterations, (2017). http://arxiv.org/abs/1706.08773.

[106] M. Tuna, A. Karthikeyan, K. Rajagopal, M. Alcin, İ. Koyuncu, Hyperjerk multiscroll oscillators with megastability: Analysis, FPGA implementation and a novel ANN-ring-based True Random Number Generator, AEU - Int. J. Electron. Commun. 112 (2019). https://doi.org/10.1016/j.aeue.2019.152941.

[107] M. Tuna, A novel secure chaos-based pseudo random number generator based on ANN-based chaotic and ring oscillator: design and its FPGA implementation, Analog Integr. Circuits Signal Process. 105 (2020) 167–181. https://doi.org/10.1007/s10470-020-01703-z.

[108] J.J.M. Chan, P. Thulasiraman, G. Thomas, R. Thulasiram, Ensuring Quality of Random Numbers

from TRNG: Design and Evaluation of Post-Processing Using Genetic Algorithm, J. Comput. Commun. 04 (2016) 73–92. https://doi.org/10.4236/jcc.2016.44007.

[109] S. Yakut, A.B. Özer, Efficient Hybrid Random Number Generator Based on Keccak, in: 2018 Int. Conf. Artif. Intell. Data Process. IDAP 2018, 2019. https://doi.org/10.1109/IDAP.2018.8620921.

[110] T. Kaya, A true random number generator based on a Chua and RO-PUF: design, implementation and statistical analysis, Analog Integr. Circuits Signal Process. 102 (2020) 415–426. https://doi.org/10.1007/s10470-019-01474-2.

[111] M. Ceyhan, E.N. Yolaçan, Görüntü Dosyalarının Şifrelenerek Güvenli Şekilde Saklanması, J. Eng. Archit. Fac. Eskisehir Osmangazi Univ. 29 (2021) 28–42. https://doi.org/10.31796/OGUMMF.825306.

[112] S. Zhou, A real-time one-time pad DNA-chaos image encryption algorithm based on multiple keys, Opt. Laser Technol. 143 (2021) 107359. https://doi.org/10.1016/j.optlastec.2021.107359.

[113] X. Wang, J. Yang, A novel image encryption scheme of dynamic S-boxes and random blocks based on spatiotemporal chaotic system, Optik (Stuttg). 217 (2020). https://doi.org/10.1016/j.ijleo.2020.164884.

[114] R. Ye, Y. Li, Y. Li, An image encryption scheme based on fractal interpolation, ACM Int. Conf. Proceeding Ser. 124 (2018) 52–56. https://doi.org/10.1145/3195588.3195596.

[115] W. Feng, Y.G. He, H.M. Li, C.L. Li, Cryptanalysis of the integrated chaotic systems based image encryption algorithm, Optik (Stuttg). 186 (2019) 449–457. https://doi.org/10.1016/j.ijleo.2018.12.103.

[116] N. Jiang, X. Dong, H. Hu, Z. Ji, W. Zhang, Quantum Image Encryption Based on Henon Mapping, Int. J. Theor. Phys. 58 (2019) 979–991. https://doi.org/10.1007/S10773-018-3989-7/FIGURES/12.

[117] T. Gao, Z. Chen, A new image encryption algorithm based on hyper-chaos, Phys. Lett. Sect. A Gen. At. Solid State Phys. 372 (2008) 394–400. https://doi.org/10.1016/j.physleta.2007.07.040.

[118] X. Chen, Y. Zhang, G. Zhang, Y. Zhang, Evaluation of ECG random number generator for wireless body sensor networks security, 2012 5th Int. Conf. Biomed. Eng. Informatics, BMEI 2012. (2012) 1308–1311. https://doi.org/10.1109/BMEI.2012.6513218.

[119] X. Chen, L. Wang, B. Li, Y. Wang, X. Li, Y. Liu, H. Yang, Modeling Random Telegraph Noise as a Randomness Source and its Application in True Random Number Generation, IEEE Trans. Comput. Des. Integr. Circuits Syst. 35 (2016) 1435–1448. https://doi.org/10.1109/TCAD.2015.2511074.

[120] M.B. Silva, Percepção da população assistida sobre a inserção de estudantes de medicina na Unidade Básica de Saúde, Trab. Conclusão Curso. 1 (2016) 1–10. https://doi.org/10.1017/CBO9781107415324.004.

[121] S.K. Mathew, S. Srinivasan, M.A. Anders, H. Kaul, S.K. Hsu, F. Sheikh, A. Agarwal, S. Satpathy, R.K. Krishnamurthy, 2.4 Gbps, 7 mW all-digital PVT-variation tolerant true random number generator for 45 nm CMOS high-performance microprocessors, IEEE J. Solid-State Circuits. 47 (2012) 2807–2821. https://doi.org/10.1109/JSSC.2012.2217631.

[122] U.C. Çabuk, Ö. Aydin, G. Dalkiliç, A random number generator for lightweight authentication protocols: Xorshiftr, Turkish J. Electr. Eng. Comput. Sci. 25 (2017) 4818–4828.

https://doi.org/10.3906/elk-1703-361.

[123] D. Yosunlu, E. Avaroğlu, Examination of Post Processing Algorithms, Bilgi. Bilim. ve Teknol. Derg. 1 (2020) 66–73.

[124] S. Ergün, Modeling and analysis of chaos-modulated dual oscillator-based random number generators, Eur. Signal Process. Conf. (2008).

[125] S. Loza, L. Matuszewski, M. Jessa, A Random Number Generator Using Ring Oscillators and SHA-256 as Post-Processing, Int. J. Electron. Telecommun. 61 (2015) 199–204. https://doi.org/10.1515/eletel-2015-0026.

[126] E. Avaroğlu, T. Tuncer, A novel S-box-based postprocessing method for true random number generation, Turkish J. Electr. Eng. Comput. Sci. 28 (2020) 288–301. https://doi.org/10.3906/elk-1906-194.

[127] J. Sen Teh, W. Teng, A. Samsudin, J. Chen, A post-processing method for true random number generators based on hyperchaos with applications in audio-based generators, Front. Comput. Sci. 14 (2020). https://doi.org/10.1007/s11704-019-9120-2.

[128] Q. Zhou, X. Liao, K. wo Wong, Y. Hu, D. Xiao, True random number generator based on mouse movement and chaotic hash function, Inf. Sci. (Ny). 179 (2009) 3442–3450. https://doi.org/10.1016/j.ins.2009.06.005.

[129] S. Burri, D. Stucki, Y. Maruyama, C. Bruschini, E. Charbon, F. Regazzoni, Jailbreak imagers: Transforming a single-photon image sensor into a true random number generator, … Image Sens. …. (2013) 5–8. http://ens.ewi.tudelft.nl/pubs/charbon13iisw_regazzoni.pdf.

[130] J. Polack, Are concert halls random number generators?, J. Acoust. Soc. Am. 120 (2006) 3101–3101. https://doi.org/10.1121/1.4787547.

[131] G. Chen, T. Ueta, Yet another chaotic attractor, Int. J. Bifurc. Chaos. 9 (1999) 1465–1466. https://doi.org/10.1142/S0218127499001024.

[132] J. Lü, G. Chen, S. Zhang, Dynamical analysis of a new chaotic attractor, Int. J. Bifurcat. Chaos. 12 (2002) 1001–1015. https://doi.org/10.1142/S0218127402004851.

[133] J.C. Sprott, Simplest dissipative chaotic flow, Phys. Lett. Sect. A Gen. At. Solid State Phys. 228 (1997) 271–274. https://doi.org/10.1016/S0375-9601(97)00088-1.

[134] G.A. Leonov, N. V. Kuznetsov, V.I. Vagaitsev, Localization of hidden Chuas attractors, Phys. Lett. Sect. A Gen. At. Solid State Phys. 375 (2011) 2230–2233. https://doi.org/10.1016/j.physleta.2011.04.037.

[135] F. Yu, L. Li, Q. Tang, S. Cai, Y. Song, Q. Xu, A Survey on True Random Number Generators Based on Chaos, Discret. Dyn. Nat. Soc. 2019 (2019). https://doi.org/10.1155/2019/2545123.

[136] H. Moqadasi, M.B. Ghaznavi-Ghoushchi, A new Chua's circuit with monolithic Chua's diode and its use for efficient true random number generation in CMOS 180 nm, Analog Integr. Circuits Signal Process. 82 (2015) 719–731. https://doi.org/10.1007/s10470-015-0498-y.

[137] E. Avaroglu, Donanım Tabanlı Rasgele Sayı Üretecinin Gerçekleştirilmesi, Fırat Üniversitesi, 2014.

[138] C.M. González, H.A. Larrondo, O.A. Rosso, Statistical complexity measure of pseudorandom bit generators, Phys. A Stat. Mech. Its Appl. 354 (2005) 281–300. https://doi.org/10.1016/j.physa.2005.02.054.

[139] J. Li, J. Zheng, P. Whitlock, Efficient deterministic and non-deterministic pseudorandom number generation, Math. Comput. Simul. 143 (2018) 114–124. https://doi.org/10.1016/j.matcom.2016.07.011.

[140] K. Marton, A. Suciu, I. Ignat, Randomness in digital cryptography: A survey, Rom. J. Inf. Sci. Technol. 13 (2010) 219–240.

[141] A. Akgul, C. Arslan, B. Aricioglu, Design of an Interface for Random Number Generators Based on Integer and Fractional Order Chaotic Systems, Chaos Teory Appl. 1 (2019) 1–18. https://dergipark.org.tr/tr/pub/chaos/653247.

[142] P. Ayubi, S. Setayeshi, A.M. Rahmani, Deterministic chaos game: A new fractal based pseudo-random number generator and its cryptographic application, J. Inf. Secur. Appl. 52 (2020) 102472. https://doi.org/10.1016/j.jisa.2020.102472.

[143] D. Lambić, M. Nikolić, New pseudo-random number generator based on improved discrete-space chaotic map, Filomat. 33 (2019) 2257–2268. https://doi.org/10.2298/FIL1908257L.

[144] A. Gholipour, S. Mirzakuchaki, A Pseudorandom Number Generator with KECCAK Hash Function, Int. J. Comput. Electr. Eng. (2011) 896–899. https://doi.org/10.7763/ijcee.2011.v3.439.

[145] J. Sen Teh, W.J. Teng, A. Samsudin, A true random number generator based on hyperchaos and digital sound, 2016 3rd Int. Conf. Comput. Inf. Sci. ICCOINS 2016 - Proc. (2016) 264–269. https://doi.org/10.1109/ICCOINS.2016.7783225.

[146] R. Morrison, Design of a True Random Number Generator Using Audio Input, J. CRAPTOLOGY. 1 (2001).

[147] I. Te Chen, J.M. Tsai, J. Tzeng, Audio random number generator and its application, Proc. - Int. Conf. Mach. Learn. Cybern. 4 (2011) 1678–1683. https://doi.org/10.1109/ICMLC.2011.6017002.

[148] Justin Salamon; Christopher Jacoby; Juan Pablo Bello, Urban Sound Datasets, MM '14 Proc. 22nd ACM Int. Conf. Multimed. (2014) 1041–1044. http://serv.cusp.nyu.edu/projects/urbansounddataset/.

[149] E. Akbal, An automated environmental sound classification methods based on statistical and textural feature, Appl. Acoust. 167 (2020) 107413. https://doi.org/10.1016/j.apacoust.2020.107413.

[150] K.J. Piczak, ESC: Dataset for environmental sound classification, MM 2015 - Proc. 2015 ACM Multimed. Conf. (2015) 1015–1018. https://doi.org/10.1145/2733373.2806390.

[151] G. Ke, H. Wang, S. Zhou, H. Zhang, Encryption of medical image with most significant bit and high capacity in piecewise linear chaos graphics, Meas. J. Int. Meas. Confed. 135 (2019) 385–391. https://doi.org/10.1016/j.measurement.2018.11.074.

[152] N. Louzzani, A. Boukabou, H. Bahi, A. Boussayoud, A novel chaos based generating function of the Chebyshev polynomials and its applications in image encryption, Chaos, Solitons and Fractals. 151 (2021) 111315. https://doi.org/10.1016/j.chaos.2021.111315.

[153] A.A. Alarood, E. Alsolami, M.A. Al-Khasawneh, N. Ababneh, W. Elmedany, IES: Hyper-chaotic plain image encryption scheme using improved shuffled confusion-diffusion, Ain Shams Eng. J. 13 (2022) 101583. https://doi.org/10.1016/J.ASEJ.2021.09.010.

[154] M. Alawida, J. Sen Teh, A. Samsudin, W.H. Alshoura, An image encryption scheme based on hybridizing digital chaos and finite state machine, Signal Processing. 164 (2019) 249–266. https://doi.org/10.1016/j.sigpro.2019.06.013.

[155] J. Fridrich, Symmetric ciphers based on two-dimensional chaotic maps, Int. J. Bifurcat. Chaos. 8 (1998) 1259–1284. https://doi.org/10.1142/S021812749800098X.

[156] K. Rama Devi, V. Janaki, G. VijayaLaxmi, Encryption methodology on crypto keys from image using chaos logistic maps, Mater. Today Proc. (2021). https://doi.org/10.1016/j.matpr.2021.01.603.

[157] J.A.P. Artiles, D.P.B. Chaves, C. Pimentel, Image encryption using block cipher and chaotic sequences, Signal Process. Image Commun. 79 (2019) 24–31. https://doi.org/10.1016/j.image.2019.08.014.

[158] D. dai Liu, W. Zhang, H. Yu, Z. liang Zhu, An image encryption scheme using self-adaptive selective permutation and inter-intra-block feedback diffusion, Signal Processing. 151 (2018) 130–143. https://doi.org/10.1016/j.sigpro.2018.05.008.

[159] X. Chai, X. Fu, Z. Gan, Y. Lu, Y. Chen, A color image cryptosystem based on dynamic DNA encryption and chaos, Signal Processing. 155 (2019) 44–62. https://doi.org/10.1016/j.sigpro.2018.09.029.

[160] C.D. Lee, B.J. Choi, K.S. Park, Design and evaluation of a block encryption algorithm using dynamic-key mechanism, Futur. Gener. Comput. Syst. 20 (2004) 327–338. https://doi.org/10.1016/S0167-739X(03)00148-1.

[161] G. Chen, Y. Mao, C.K. Chui, A symmetric image encryption scheme based on 3D chaotic cat maps, Chaos, Solitons and Fractals. 21 (2004) 749–761. https://doi.org/10.1016/j.chaos.2003.12.022.

[162] A. Singh, P. Agarwal, M. Chand, Image Encryption and Analysis using Dynamic AES, 2019 Int. Conf. Optim. Appl. ICOA 2019. (2019) 1–6. https://doi.org/10.1109/ICOA.2019.8727711.

[163] A. Akhshani, S. Behnia, A. Akhavan, H. Abu Hassan, Z. Hassan, A novel scheme for image encryption based on 2D piecewise chaotic maps, Opt. Commun. 283 (2010) 3259–3266. https://doi.org/10.1016/j.optcom.2010.04.056.

[164] S. Dhall, S.K. Pal, K. Sharma, A chaos-based probabilistic block cipher for image encryption, J. King Saud Univ. - Comput. Inf. Sci. 34 (2022) 1533–1543. https://doi.org/10.1016/j.jksuci.2018.09.015.

[165] I.C. Sari, M. Zarlis, T. Tulus, Optimization of Data Encryption Modeling Using RSA Cryptography Algorithm As Security E-Mail Data, J. Phys. Conf. Ser. 1471 (2020). https://doi.org/10.1088/1742-6596/1471/1/012068.

[166] Q. Yu, C.N. Zhang, A new and fast cryptographic hash function based on RC4, Cryptologia. 40 (2016) 522–540. https://doi.org/10.1080/01611194.2015.1135486.

[167] Y. Hu, Y. Wang, J. Zhao, Z. Tang, X. Gong, J. Gao, Design of nonlinear hierarchical controller for intake manifold pressure and boost pressure of turbocharged gasoline engine, Int. J. Veh. Des. 82 (2020) 161. https://doi.org/10.1504/ijvd.2020.10036464.

[168] M. Asgari-Chenaghlu, M.A. Balafar, M.R. Feizi-Derakhshi, A novel image encryption algorithm based on polynomial combination of chaotic maps and dynamic function generation, Signal Processing. 157 (2019) 1–13. https://doi.org/10.1016/j.sigpro.2018.11.010.

[169] F.G. Jeng, W.L. Huang, T.H. Chen, Cryptanalysis and improvement of two hyper-chaos-based image encryption schemes, Signal Process. Image Commun. 34 (2015) 45–51. https://doi.org/10.1016/j.image.2015.03.003.

[170] G. Álvarez, F. Montoya, M. Romera, G. Pastor, Cryptanalysis of a discrete chaotic cryptosystem

using external key, Phys. Lett. Sect. A Gen. At. Solid State Phys. 319 (2003) 334–339. https://doi.org/10.1016/j.physleta.2003.10.044.

[171] Y. Zhang, The unified image encryption algorithm based on chaos and cubic S-Box, Inf. Sci. (Ny). 450 (2018) 361–377. https://doi.org/10.1016/j.ins.2018.03.055.

[172] A. Kadir, M. Aili, M. Sattar, Color image encryption scheme using coupled hyper chaotic system with multiple impulse injections, Optik (Stuttg). 129 (2017) 231–238. https://doi.org/10.1016/j.ijleo.2016.10.036.

[173] H. Zhao, S. Xie, J. Zhang, T. Wu, A dynamic block image encryption using variable-length secret key and modified Henon map, Optik (Stuttg). 230 (2021) 166307. https://doi.org/10.1016/j.ijleo.2021.166307.

[174] B. Stoyanov, K. Kordov, Image encryption using chebyshev map and rotation equation, Entropy. 17 (2015) 2117–2139. https://doi.org/10.3390/E17042117.

[175] W. Liu, K. Sun, C. Zhu, A fast image encryption algorithm based on chaotic map, Opt. Lasers Eng. 84 (2016) 26–36. https://doi.org/10.1016/j.optlaseng.2016.03.019.

[176] J. Wu, X. Liao, B. Yang, Color image encryption based on chaotic systems and elliptic curve ElGamal scheme, Signal Processing. 141 (2017) 109–124. https://doi.org/10.1016/j.sigpro.2017.04.006.

[177] V.M. Manikandan, A.A. Bini, An Improved Reversible Data Hiding Through Encryption Scheme with Block Prechecking, Procedia Comput. Sci. 171 (2020) 951–958. https://doi.org/10.1016/j.procs.2020.04.103.

[178] M. Shrivastava, S. Roy, K. Kumar, C.V. Pandey, J. Grover, LICCA: a lightweight image cipher using 3-D cellular automata, Nonlinear Dyn. 106 (2021) 2679–2702. https://doi.org/10.1007/s11071-021-06923-0.

[179] Z. Man, J. Li, X. Di, Y. Sheng, Z. Liu, Double image encryption algorithm based on neural network and chaos, Chaos, Solitons and Fractals. 152 (2021) 111318. https://doi.org/10.1016/j.chaos.2021.111318.

[180] F. Ikbal, R. Gopikakumari, Image block generation from block-based SMRT in colour image encryption and its performance analysis, J. King Saud Univ. - Comput. Inf. Sci. (2021). https://doi.org/10.1016/J.JKSUCI.2021.08.026.

[181] M. Shi, S. Guo, X. Song, Y. Zhou, E. Wang, Visual secure image encryption scheme based on compressed sensing and regional energy, Entropy. 23 (2021). https://doi.org/10.3390/E23050570.

[182] D.K.R. Shukla, V.K.R. Dwivedi, M.C. Trivedi, Encryption algorithm in cloud computing, Mater. Today Proc. 37 (2021) 1869–1875. https://doi.org/10.1016/J.MATPR.2020.07.452.

[183] K.K. Raghuvanshi, S. Kumar, S. Kumar, S. Kumar, Development of new encryption system using Brownian motion based diffusion, Multimed. Tools Appl. (2021). https://doi.org/10.1007/s11042-021-10665-x.

[184] Y. Ma, C. Li, B. Ou, Cryptanalysis of an image block encryption algorithm based on chaotic maps, J. Inf. Secur. Appl. 54 (2020) 102566. https://doi.org/10.1016/J.JISA.2020.102566.

[185] H. Liu, A. Kadir, Y. Niu, Chaos-based color image block encryption scheme using S-box, AEU - Int. J. Electron. Commun. 68 (2014) 676–686. https://doi.org/10.1016/J.AEUE.2014.02.002.

[186] A. Devi, A. Sharma, A. Rangra, A Review on DES, AES and Blowfish for Image Encryption & Decryption, Int. J. Eng. Comput. Sci. 4 (2015) 12646–12651. https://www.ijecs.in/index.php/ijecs/article/download/3887/3623/.

[187] P. Refregier, B. Javidi, Optical image encryption based on input plane and Fourier plane random encoding, Opt. Lett. 20 (1995) 767. https://doi.org/10.1364/ol.20.000767.

[188] A.M. Garipcan, E. Erdem, T. Tuncer, Donanım Tabanlı Trivium Akış Şifreleme Algoritmasının FPGA Ortamında Gerçekleştirilmesi, Fırat Üniv. Müh. Bil. Derg. 29 (2017) 119–130.

[189] G. Bertoni, Keccak sponge function family main document, Nist. (2009) 1–121.

[190] www.drdobbs.com, Keccak: The New SHA-3 Encryption Standard, Dr. Dobbs. 3 (2019) 1–7. http://www.drdobbs.com/security/keccak-the-new-sha-3-encryption-standard/240154037 (accessed March 10, 2022).

[191] G. Bertoni, J. Daemen, S. Hoffert, M. Peeters, G. Van Assche, R. Van Keer, Farfalle : parallel permutation-based cryptography, (2018) 1–37.

[192] H. Zhou, R. Zong, X. Dong, K. Jia, W. Meier, Interpolation Attacks on Round-Reduced Elephant, Kravatte and Xoofff, Comput. J. 64 (2021) 628–638. https://doi.org/10.1093/comjnl/bxaa101.

[193] B. Ritz, P.C. Lee, C.F. Lassen, O.A. Arah, Parkinson disease and smoking revisited: Ease of quitting is an early sign of the disease, 2014. https://doi.org/10.1212/WNL.0000000000000879.

[194] U.M. Maurer, A universal statistical test random bit generators, Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 537 LNCS (1991) 409–420. https://doi.org/10.1007/3-540-38424-3_30.

[195] E. Avaroğlu, A.B. Özer, M. Türk, The study of the reasons for not giving the results of NIST Tests of Random Walk, Random Walk Variable and Lempel Ziv, Turkish J. Sci. Technol. 10 (2015) 1–8. https://dergipark.org.tr/en/pub/tjst/issue/29285/313541 (accessed March 14, 2022).

[196] L. Vinet, A. Zhedanov, A "missing" family of classical orthogonal polynomials, 2011. https://doi.org/10.1088/1751-8113/44/8/085201.

[197] A.J. Menezes, P.C. Van Oorschot, S.A. Vanstone, Handbook of applied cryptography, Handb. Appl. Cryptogr. (1996) 1–780. https://doi.org/10.2307/2589608.

[198] S. Kodba, M. Perc, M. Marhl, Detecting chaos from a time series, Eur. J. Phys. 26 (2005) 205–215. https://doi.org/10.1088/0143-0807/26/1/021.

[199] H. Kevs, Ö. Mutlu, O.H. Koçal, Lyapunov Üstelleri İle İris Örüntüsünün Kaotik Yapısının İncelenmesi Investigation Of Chaotic Structure Of Iris Pattern By Using Lyapunov Exponents, 5 (2017) 405–408.

[200] K. Binder, Monte Carlo Simulations in Statistical Physics, 3rd ed., Cambridge University Press, Cambridge, 2017. https://doi.org/10.1007/978-3-642-27737-5_337-2.

[201] J. Hoshen, R. Kopelman, Percolation and cluster distribution. I. Cluster multiple labeling technique and critical concentration algorithm, Phys. Rev. B. 14 (1976) 3438–3445. https://doi.org/10.1103/PhysRevB.14.3438.

[202] F. Özkaynak, A.B. Özer, S. Yavuz, Kaos Tabanlı Yeni Bir Blok Şifreleme Algoritması, in: IV.Ağ ve Bilgi Güvenliği Ulus. Sempozyumu, 2011: pp. 1–6.

[203] J.S. Khan, J. Ahmad, M.A. Khan, TD-ERCS map-based confusion and diffusion of autocorrelated

data, Nonlinear Dyn. 87 (2017) 93–107. https://doi.org/10.1007/s11071-016-3028-2.

[204] E. Avaroğlu, İ. Koyuncu, A.B. Özer, M. Türk, Hybrid pseudo-random number generator for cryptographic systems, Nonlinear Dyn. 82 (2015) 239–248. https://doi.org/10.1007/s11071-015-2152-8.

[205] Y. Xian, X. Wang, X. Yan, Q. Li, X. Wang, Image Encryption Based on Chaotic Sub-Block Scrambling and Chaotic Digit Selection Diffusion, Opt. Lasers Eng. 134 (2020) 106202. https://doi.org/10.1016/j.optlaseng.2020.106202.

[206] G. Dursun, F. Özer, U. Özkaya, A new and secure digital image scrambling algorithm based on 2D cellular automata, Turkish J. Electr. Eng. Comput. Sci. 25 (2017) 3515–3527. https://doi.org/10.3906/elk-1610-225.

[207] H.S. Ye, N.R. Zhou, L.H. Gong, Multi-image compression-encryption scheme based on quaternion discrete fractional Hartley transform and improved pixel adaptive diffusion, Signal Processing. 175 (2020). https://doi.org/10.1016/j.sigpro.2020.107652.

[208] X. Wang, J. Yang, A novel image encryption scheme of dynamic S-boxes and random blocks based on spatiotemporal chaotic system, Optik (Stuttg). 217 (2020) 164884. https://doi.org/10.1016/j.ijleo.2020.164884.

[209] Y. Sang, J. Sang, M.S. Alam, Image encryption based on logistic chaotic systems and deep autoencoder, Pattern Recognit. Lett. 153 (2022) 59–66. https://doi.org/10.1016/j.patrec.2021.11.025.

[210] A. Karcı, F. Bilgiç, Karcı ve Shannon Entropilerin Karşılaştırılması, Anatol. J. Comput. Sci. 4 (2019) 68–73.

[211] O. Bahadır, H. Türkmençalıkoğlu, Bilgi Kuramında Shannon Entropisi ve Uygulamaları, Eur. J. Sci. Technol. 32 (2021) 491–497. https://doi.org/10.31590/ejosat.1039771.

[212] Y. Wu, J.P. Noonan, S. Agaian, NPCR and UACI Randomness Tests for Image Encryption, J. Sel. Areas Telecommun. April Edit (2011) 31–38. http://www.cyberjournals.com/Papers/Apr2011/05.pdf.

[213] X. Liu, Y. Wu, H. Zhang, J. Wu, L. Zhang, Quaternion discrete fractional Krawtchouk transform and its application in color image encryption and watermarking, Signal Processing. 189 (2021) 108275. https://doi.org/10.1016/j.sigpro.2021.108275.

[214] T. Etem, T. Kaya, Self-generated encryption model of acoustics, Appl. Acoust. 170 (2020) 107481. https://doi.org/10.1016/j.apacoust.2020.107481.

[215] M. Alawida, A. Samsudin, J. Sen Teh, R.S. Alkhawaldeh, A new hybrid digital chaotic system with applications in image encryption, Signal Processing. 160 (2019) 45–58. https://doi.org/10.1016/j.sigpro.2019.02.016.

[216] J. Chen, Z. liang Zhu, L. bo Zhang, Y. Zhang, B. qiang Yang, Exploiting self-adaptive permutation–diffusion and DNA random encoding for secure and efficient image encryption, Signal Processing. 142 (2018) 340–353. https://doi.org/10.1016/j.sigpro.2017.07.034.

[217] H. Wang, D. Xiao, X. Chen, H. Huang, Cryptanalysis and enhancements of image encryption using combination of the 1D chaotic map, Signal Processing. 144 (2018) 444–452. https://doi.org/10.1016/j.sigpro.2017.11.005.

[218] J. Wu, X. Liao, B. Yang, Image encryption using 2D Hénon-Sine map and DNA approach, Signal

Processing. 153 (2018) 11–23. https://doi.org/10.1016/j.sigpro.2018.06.008.

[219] C. Pak, L. Huang, A new color image encryption using combination of the 1D chaotic map, Signal Processing. 138 (2017) 129–137. https://doi.org/10.1016/j.sigpro.2017.03.011.

[220] K. Ratnavelu, M. Kalpana, P. Balasubramaniam, K. Wong, P. Raveendran, Image encryption method based on chaotic fuzzy cellular neural networks, Signal Processing. 140 (2017) 87–96. https://doi.org/10.1016/j.sigpro.2017.05.002.

[221] A. Belazi, A.A. Abd El-Latif, S. Belghith, A novel image encryption scheme based on substitution-permutation network and chaos, Signal Processing. 128 (2016) 155–170. https://doi.org/10.1016/j.sigpro.2016.03.021.

[222] C. Cao, K. Sun, W. Liu, A novel bit-level image encryption algorithm based on 2D-LICM hyperchaotic map, Signal Processing. 143 (2018) 122–133. https://doi.org/10.1016/j.sigpro.2017.08.020.

[223] M. Ghebleh, A. Kanso, A novel efficient image encryption scheme based on chained skew tent maps, Neural Comput. Appl. 31 (2019) 2415–2430. https://doi.org/10.1007/s00521-017-3199-x.

[224] L. Xu, X. Gou, Z. Li, J. Li, A novel chaotic image encryption algorithm using block scrambling and dynamic index based diffusion, Opt. Lasers Eng. 91 (2017) 41–52. https://doi.org/10.1016/j.optlaseng.2016.10.012.

[225] X. Chai, Z. Gan, M. Zhang, A fast chaos-based image encryption scheme with a novel plain image-related swapping block permutation and block diffusion, Multimed. Tools Appl. 76 (2017) 15561–15585. https://doi.org/10.1007/s11042-016-3858-4.

[226] L. Huang, S. Cai, X. Xiong, M. Xiao, On symmetric color image encryption system with permutation-diffusion simultaneous operation, Opt. Lasers Eng. 115 (2019) 7–20. https://doi.org/10.1016/j.optlaseng.2018.11.015.

[227] C. Zhu, A novel image encryption scheme based on improved hyperchaotic sequences, Opt. Commun. 285 (2012) 29–37. https://doi.org/10.1016/j.optcom.2011.08.079.

[228] M. Khan, T. Shah, An efficient chaotic image encryption scheme, Neural Comput. Appl. 26 (2015) 1137–1148. https://doi.org/10.1007/s00521-014-1800-0.

ÖZGEÇMİŞ

Taha ETEM

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

AKADEMİK FAALİYETLER

Makaleler:

1. T. Etem, T. Kaya, A novel True Random Bit Generator design for image encryption, Phys. A Stat. Mech. Its Appl. 540 (2020). https://doi.org/10.1016/j.physa.2019.122750.
2. T. Etem, T. Kaya, Self-generated encryption model of acoustics, Appl. Acoust. 170 (2020) 107481. https://doi.org/10.1016/j.apacoust.2020.107481.
3. T. Etem, T. Kaya, Görüntü Şifreleme için Trivium-Doğrusal Eşlenik Üreteci Tabanlı Bit Üretimi, Fırat Üniversitesi Mühendislik Bilim. Derg. 32 (2020) 287–294. https://doi.org/10.35234/fumbd.687403.
4. T. Etem, T. Kaya, Trivium Algoritması Kaynaklı Rastgele Permutasyon Üretimiyle Görüntü Şifreleme Uygulaması, Fırat Üniversitesi Mühendislik Bilim. Derg. 34 (2022) 687-697. https://doi.org/10.35234/fumbd.1122197.

5. T. Etem, T. Abbasov, Measurements Of High Frequency Electromagnetic Waves In Center Of Mus, International Journal of Applied Mathematics, Electronics and Computers (2016) 95-97.

Bildiriler:

1. T. Etem, T. Kaya, A New Encryption Key Generator Design with True Random Bits, ISATECH '21 Syposium, 28-30 June 2021.
2. T. Etem, T. Kaya, Gerçek Veri ile Paket Program Kaynaklarından Gerçekleştirilen Rastgele Sayı Üreteçlerinin Karşılaştırılması, ICIAS 1st Conference '22 , 2022.
3. T. Etem, T. Karadağ, T. Abbasov, Mapping Mobile Phone Based Electromagnetic Pollution in Mus City Centre, ICAIE '17 Conference, 490-493, 2017.
4. İ. Bozkurt, Z. Pala, T. Etem, Measurement and evaluation of low frequency electromagnetic field in camera observation rooms, 2017 XIIIth International Conference on Perspective Technologies and Methods in MEMS Design (MEMSTECH), 2017.
5. Z. Pala, İ. Bozkurt, T. Etem, Estimation of low frequency electromagnetic values using machine learning, 2017 XIIIth International Conference on Perspective Technologies and Methods in MEMS Design (MEMSTECH), 2017.
6. T. Etem, Z. Pala, İ. Bozkurt, Electromagnetic pollution measurement in the system rooms of a university, 2017 XIIIth International Conference on Perspective Technologies and Methods in MEMS Design (MEMSTECH), 2017.
7. T. Etem, T. Abbasov, Seçilmis Bir Bölgede Yeni Nesil Iletisim Teknolojilerinin Yaydıgı Elektromanyetik Alan Seviyesinin Ölçümü ve Degerlendirilmesi, IDAP '16, 2016.
8. T. Etem, T. Abbasov, Measurements of High Frequency Electromagnetic Waves in Center of Mus, ICAT '16, 731-733, 2016.
9. T. Etem, T. Abbasov, Electric field measurements of a base station at 2G and 3G frequencies, 2016 XII International Conference on Perspective Technologies and Methods in MEMS Design (MEMSTECH), 2016.
10. M. Çeçen, Z. Pala, T. Etem, Elektromanyetık Alanların Bulanık Mantık Yardımıyla Saglık Üzerıne Etkı Derecesının Belırlenmesi, ICNASE '16, 341-346, 2016.
11. T. Etem, Z. Pala, M. Çeçen, Bir Üniversite Kampüsünde Elektromanyetik Kirliligin Zaman Içindeki Degisiminin Arastırılması, ICNASE '16, 426-432, 2016
12. T. Etem, T. Abbasov, Yasam Alanları Içerisinde GSM Bazlı Elektromanyetik Kirliligin Gün Içindeki Degisiminin Incelenmesinde Mus Örnegi, Akademik Bilişim '16, 2016.
13. Z. Pala, T. Etem, M. Çeçen, Mus Alparslan Üniversitesi Kampüsünde Elektromanyetik Radyasyon Degerlerinin Ölçülmesi ve Risk Alanlarının Belirlenmesi, EMANET '15, 2015.

Projeler:

1. Mus Alparslan Üniversitesi Kampüsünde Elektromanyetik Risk Unsurlarının Belirlenmesi, MSÜ15 MMF G01, Muş Alparslan Üniversitesi BAP Birimi, Araştırmacı: Taha Etem.
2. Mus Alparslan Üniversitesi Yerleskelerinde Elektrik ile Çalısan Cihazların Olusturdugu Düsük Frekanslı Elektromanyetik Alanın Ölçülmesi ve Etkilerinin Degerlendirilmesi, MSÜ16 MMF G01, Muş Alparslan Üniversitesi BAP Birimi, Araştırmacı: Taha Etem.
3. Mus Bölgesinde ve Çevresinde Elektromanyetik Alan Yayılmasının Incelenmesi ve Degerlendirilmesi, İnönü Üniversitesi BAP Birimi, Araştırmacı: Taha Etem.