T.C.
BURDUR MEHMET AKİF ERSOY ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
YÜKSEK LİSANS TEZİ

ALGILAYICI TABANLI GERÇEK RASTGELE SAYI ÜRETECİ TASARIMI VE ANALİZİ

Oğuzhan ARSLAN

Danışman: Doç. Dr. İsmail KIRBAŞ

BURDUR, 2023

ETİK KURALLARA UYGUNLUK BEYANI

Burdur Mehmet Akif Ersoy Üniversitesi Fen Bilimleri Enstitüsü Lisansüstü Eğitim ve Öğretim Yönetmeliğinin ilgili hükümleri uyarınca Yüksek Lisans Tezi olarak sunduğum **"Algılayıcı Tabanlı Gerçek Rastgele Sayı Üreteci Tasarımı ve Analizi"** başlıklı bu tezin;

- Kendi çalışmam olduğunu,
- Sunduğum tüm sonuç, doküman, bilgi ve belgeleri bizzat ve bu tez çalışması kapsamında elde ettiğimi,
- Bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara atıf yaptığımı ve bunları kaynaklar listesinde usulüne uygun olarak verdiğimi,
- Kullandığım verilerde değişiklik yapmadığımı,
- Tez çalışması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını,
- Bu tezin herhangi bir bölümünü bu üniversite veya diğer bir üniversitede başka bir tez çalışması içinde sunmadığımı,
- Bu tezin planlanmasından yazımına kadar bütün safhalarda bilimsel etik kurallarına uygun olarak davrandığımı,

bildirir, aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul edeceğimi beyan ederim.

03/07/2023

Oğuzhan ARSLAN

ÖNSÖZ

Bu araştırma için beni yönlendiren, karşılaştığım zorlukları bilgi ve tecrübesi ile aşmamda yardımcı olan değerli hocam Doç. Dr. İsmail KIRBAŞ'a teşekkürlerimi sunarım.

Araştırmalarımın her anında yanımda olan sevgili eşim Eda'ya teşekkür ederim.

2021-YL-0800 No'lu Proje ile tezimi maddi olarak destekleyen Burdur Mehmet Akif Ersoy Üniversitesi Bilimsel Araştırma Projeleri Koordinatörlüğü'ne teşekkür ederim.

Eğitim hayatımın her aşamasında beni her anlamda destekleyen aileme sonsuz sevgi ve saygılarımı sunarım.

Temmuz, 2023 **Oğuzhan ARSLAN**

İÇİNDEKİLER

Sayfa

ŞEKİLLER DİZİNİ

Sayfa

ÇİZELGELER DİZİNİ

Sayfa

SİMGELER VE KISALTMALAR DİZİNİ

3DES : Üçlü Veri Şifreleme Standardı

AES : Gelişmiş Şifreleme Standardı

CMOS : Bütünleyici Metal Oksit Yarı İletken

DES : Veri Şifreleme Standardı

FIPS : Federal Bilgi İşleme Standartları

GND : Toprak

GPS **:** Küresel Konumlama Sistemi

GRSÜ **:** Gerçek Rastgele Sayı Üreteçleri

GSM : Mobil İletişim İçin Küresel Sistem

IBM : Uluslararası İş Makineleri

IOT **:** Nesnelerin İnterneti

JSON : JavaScript Nesnesi Gösterimi

LCG : Doğrusal Uyumlu Üreteç

LDR : Işığa Bağımlı Direnç

MD5 : Mesaj Özeti 5

NIST : Ulusal Standartlar ve Teknoloji Enstitüsü

OpenCV : Açık Kaynak Bilgisayar Görüsü

RSA : Rivest, Shamir ve Adleman

RSÜ **:** Rastgele Sayı Üreteçleri

SCL : Seri Saat

SDA : Seri Veri

SHA : Güvenli Özetleme Algoritması

SRSÜ **:** Sözde Rastgele Sayı Üreteçleri

USB : Evrensel Seri Veriyolu

ÖZET

Yüksek Lisans Tezi

Algılayıcı Tabanlı Gerçek Rastgele Sayı Üreteci Tasarımı ve Analizi

Oğuzhan ARSLAN

Burdur Mehmet Akif Ersoy Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. İsmail KIRBAŞ

Temmuz, 2023

Birçok alanda, belirsizliğin ve öngörülemezliğin rastgele sayılar aracılığıyla temsil edilmesi çok önemlidir. Bu sayılar bilgisayar bilimi, kriptografi ve istatistik gibi sistemlerin ve prosedürlerin güvenliğini sağlamaya yardımcı olduğu alanlarda hayati önem taşır. Şifreler, anahtarlar ve diğer güvenlik simgelerini oluşturmanın yanı sıra algoritmalara ve simülasyonlara rastgelelik katmak için bilgisayar bilimciler rastgele sayı üretiminden yararlanır. Yakın zamanda yapılan çalışmalar, milyarlarca IoT cihazında bulunan donanımsal rastgele sayı üreteçleri tarafından üretilen entropinin yetersiz olduğunu ortaya koymuştur. Bu araştırma, IoT sensör verilerinin kriptografi sistemleri için rastgele sayılar üretmek üzere nasıl kullanılabileceğini açıklamanın yanı sıra, bu rastgele sayıların etkilerini de incelemektedir. Kamera, ışık ve renk sensörleri, algılayıcı olarak kullanılmıştır. Öngörülemezlik seviyesini ölçmek için monobit ve poker testlerinden elde edilen sonuçlar değerlendirilmiştir. Analiz sonucunda kaliteli değerler üreten teknik kullanılarak bir milyon ve yirmi bin bit uzunluğunda diziler oluşturulmuş ve bu diziler daha sonra NIST ve FIPS 140-1 rastgelelik testi yazılım paketlerine girilmiştir. Bu iki testin sonuçları incelendiğinde dizilerin her bir testi başarıyla geçtiği görülmüştür.

Anahtar Kelimeler: kriptografi, nesnelerin interneti, rastgele sayı üreteçleri, ışık sensörü, webcam sensörü

Hazırlanan bu Yüksek Lisans tezi Burdur Mehmet Akif Ersoy Üniversitesi Bilimsel Araştırma Projeleri Komisyonu tarafından 0800-YL-21 proje numarası ile desteklenmiştir.

ABSTRACT

M.Sc. Thesis

Design and Analysis of a Sensor Based True Random Number Generator

Oğuzhan ARSLAN

Burdur Mehmet Akif Ersoy University
Graduate School of Natural and Applied Sciences
Computer Engineering Department

Supervisor: Assoc. Prof. Dr. İsmail KIRBAŞ

July, 2023

In many fields, representing uncertainty and unpredictability through random numbers is crucial. These numbers are vital in fields such as computer science, cryptography and statistics where they help to secure systems and procedures. Computer scientists use random number generation to generate passwords, keys and other security tokens, as well as to introduce randomness into algorithms and simulations. Recent studies have shown that the entropy generated by hardware random number generators in billions of IoT devices is insufficient. This research not only describes how IoT sensor data can be used to generate random numbers for cryptography systems, but also examines the effects of these random numbers. Camera, light and colour sensors are used as sensors. Results from monobit and poker tests are analysed to measure the level of unpredictability. As a result of the analysis, sequences of one million and twenty thousand bits in length were created using the technique that produces quality values, and these sequences were then entered into the NIST and FIPS 140-1 randomness test software packages. When the results of these two tests were analysed, it was seen that the sequences passed each test successfully.

Keywords: cryptography, internet of things, , random number generators, light sensor, webcam sensor

The present M.Sc. Thesis was supported by Burdur Mehmet Akif Ersoy University Scientific Research Projects CommissionUnder the Project number of 0800-YL-21.

1. GİRİŞ

İnsan hayatında çok önemli bir yere sahip olan teknoloji; her geçen gün gelişmekte ve vazgeçilmez bir yapı haline gelmektedir. Günümüzde teknoloji, başlı başına bir kavram olmaktan çıkmış ve nesneleri ilgilendiren IoT kavramının ortaya çıkmasına katkı sağlamıştır. Son yıllarda sağlık, tarım, hayvancılık, enerji gibi (Al Salami vd., 2016) hayatımızın birçok bölümüne giren IoT sistemleri ile insan müdahalesi olmadan (Naru vd., 2017) cihazları uzaktan yönetme, kontrol etme, veri aktarabilme gibi yeteneklerin yaşantımıza kazandırılması hedeflenmektedir. Bu yetenekler kazandırılırken; yeni ve gelişmekte olan bir teknoloji olması sebebiyle; doğru teknoloji seçimi, hizmet sağlayıcı ve alıcıların birbirleriyle uyumlu şekilde haberleşmesi, maliyet ve iletişim güvenliği gibi aşılması gereken bazı problemler ve zorluklar da ortaya çıkmıştır. Bir sanayi devrimi olarak ortaya çıkan Endüstri 4.0'dan farklı olarak Toplum 5.0 kavramı; büyük veri, robot ve yapay zekâ gibi yenilikleri hem sanayi de hem de sosyal hayatta kullanarak bir takım sosyal problemleri çözmeyi hedeflemektedir. Bu kavram, siber dünya ve fiziksel alanın birbirine uyumlu bir şekilde entegre edilmesiyle, toplumun mutlu ve refah içinde yaşamasını sağlamaktadır. Japonya tarafından ortaya konan ve bir başka ifadeyle "süper akıllı toplum" olarak da isimlendirilen Toplum 5.0, insan merkezli bir vizyona sahiptir (Saracel ve Aksoy, 2020). Bir dünya projesi olan Toplum 5.0'ın temelinde, tek merkezi reddedip dağıtık sistemi destekleyen bir felsefe vardır.

Bilişim sistemleri konusundaki gelişmelerin hızla ilerlemesi sonucu haberleşme sistemleri, akıllı mobil cihazlar günümüzün ayrılmaz bir parçası haline gelmiştir. Telekomünikasyon hizmetleri üzerinden gerçekleşen çevrimiçi bankacılık hizmetlerinden kritik haberleşme altyapılarına, cep telefonlarından elektronik posta sistemlerine kadar birçok bilgi alışverişi gerektiren sistemdeki kullanıcı sayısı her geçen gün artmaktadır. Artan bu kullanıcı sayısı sonucunda çağlar boyunca tarihin akışını şekillendiren ve insanlar için her zaman kıymetli olan bilginin gelebilecek tehditlere karşı mutlaka korunması ihtiyacı ortaya çıkmıştır. Uç cihazların artması, gizlilik ve siber güvenlik tehditlerini de artırmaktadır (Dash ve Choudekar, 2021). Söz konusu tehditlerin azaltılması için iletişim altyapısının güvenli bir şekilde tasarlanması, tehdidin uç cihaza gelmeden engellenmesini sağlayacak önlemler alınmalıdır.

Büyük çaplı altyapılara sahip olan haberleşme ağları üzerindeki veri alışverişinin kötü niyetli kişiler tarafından gelecek saldırılara karşı güvenliğinin alınabilmesi için gizlilik, bütünlük ve kimlik denetimi ihtiyaçlarını sağlaması gerekmektedir. Teknolojinin

çok fazla gelişmediği eski dönemlerde ilkel yöntemler ile bu ihtiyaç sağlanırken günümüzde çok farklı ve gelişmiş mekanizmalar ile saldırılara karşı savunma gereksinimi sağlanmaktadır. Savunma mekanizmalarının gelişmesiyle beraber saldırı tekniklerinde de gelişmeler olması ve bilgisayarların hesaplama gücünün artması, her geçen gün karmaşıklığı artan haberleşme sistemlerindeki güvenlik risklerinde artışlar olmasına sebep olmuştur. Böylece veri alışverişini tehlikeye atan yeni tehditlerin ortaya çıkması ve güvenlik açıklarının oluşması, savunma yapılarında daha farklı yöntemlerin geliştirilmesi zorunluluğunu ortaya çıkarmıştır.

Günümüz haberleşme sistemlerinde veri alışverişinin güvenliğini yüksek seviyede sağlayan, verilere yetkisiz olan erişimleri engellemek amacıyla ihtiyaç duyulan güvenlik politikalarının tasarımıyla ilgilenen kavrama kriptografi denir. Modern kriptografi mekanizmaları, saldırganlar tarafından bilinen anahtar tabanlı birer algoritmadır. Bu algoritmalar güvenlik ve gizliliği sağlamak amacıyla açık anahtar, kapalı anahtar, başlangıç vektörü vb. rastgele üretilmiş sayılara gereksinim duymaktadır. Bu özel ve gizli anahtarlar sistemin içerisinde veya dışarısında gömülü olarak üretilebilmektedir. Oluşabilecek tehditlere karşı güvenlik, algoritmanın karmaşıklığı yerine yüksek güvenliğe sahip rastgele sayılar ile sağlanmaktadır. Kriptografik gereksinimleri karşılamayan ya da istenilen yöntemlerle elde edilmeyen rastgele sayılar, deterministik sonuçlar doğurarak haberleşme mekanizmasının güçsüz olmasına sebep olacaktır.

Rastgele sayılar, sıfır veya bir rakamlarından oluşan, belli bir uzunluğa sahip değerleri içerisindeki elemanlarının oluşma olasılığı eşit olan ve rakamları arasında herhangi bir ilişki bulunmayan sayı dizelerinden oluşur. Üretilen sayıların istenmeyen kişiler tarafından tahmin edilebilmesi, kabul edilebilir bir durum olmadığından dolayı rastgele sayıların üretiminde tahmin edilememe özelliği hayati bir önem taşımaktadır. Kriptografi haricinde bilgisayar oyunları, piyango, istatistik gibi alanlarda da kullanılan rastgele sayılar kritik bir unsur haline gelmiştir.

Rastgele Sayı Üreteçleri (RSÜ), yazılımsal veya donanımsal olarak üretilebilen, birbirini tekrar etmeyen dizelerden oluşan ve gelecekte üretilecek sayıların tahmin edilemediği bir güvenlik mekanizmasıdır. RSÜ'ler, Sözde Rastgele Sayı Üreteçleri (SRSÜ) ve Gerçek Rastgele Sayı Üreteçleri (GRSÜ) olmak üzere iki farklı mimaride tasarlanmaktadır. Sözde Rastgele Sayı Üreteçleri, uygulamasının kolay olması ve üretim maliyetinin düşük olması sebebiyle daha fazla tercih edilmektedir. Herhangi bir donanım olmadan rastgele sayıların hızlı bir şekilde üretilmesi bilgisayar sistemlerinde büyük bir avantajdır. Gerçek Rastgele Sayı Üreteçleri ise gürültü kaynağı olarak deterministik

olmayan sayıları içermektedir ve güvenli haberleşme platformları için çok büyük önem taşımaktadır. Çevrede var olan gerçek rastgeleliği yakalamak maksadıyla pahalı donanımlar gerekir. Üretilecek rastgele sayıların kuvvetli istatistikler göstermesi, tahmin edilemez olması, düzgün bir yapıya sahip olması ve Sözde RSÜ'lerden farklı olarak donanımların kullanılması gizlilik açısından çok önem arz etmektedir. Üretilen sayıların yüksek güvenlik gerektiren kriptografik sistemlerde kullanılması durumunda bazı matematiksel gerekliliklerin (rastgelelik testleri) karşılanması gerekmektedir. Bu gerekliliklerin olumlu sonuçlanması durumunda kritik sistemlerde kullanılan algoritmalar için anahtar değeri olarak kullanım onayı alınmış olur.

1.1. Tezin Amacı ve Önemi

1990'lı yıllara gelindiğinde son derece hızlı bir şekilde ilerleyen teknoloji alanında siber uzay kavramı ortaya çıkmıştır. Siber uzay, haberleşme alt yapı sistemleriyle birbirine bağlanan elektronik sistemlerin bilgiyi paylaştığı alana verilen isimdir. NATO tarafından kara, deniz ve havadan sonra dördüncü savaş alanı olarak nitelendirilen siber uzay, devletlerin güvenliğini ciddi derecede tehdit eden bir unsur haline gelmiştir. Siber ordularını kurmaya başlayan Çin, ABD ve Rusya gibi süper güçler, düşman olduğu güçlere büyük ölçüde zararlar vermektedir. Türkiye, 25 milyon saldırı ile ABD ve Brezilya'nın ardından en fazla siber saldırıya uğrayan 3. ülke konumundadır. Günümüzde yaklaşık 4 milyar insan internet kullanmakta, yılda yaklaşık 200 milyar elektronik posta alışverişi yapılmaktadır. Bu istatistiklere bakılacak olursa siber dünya, insanlar için ayrılmaz bir parça haline gelmiştir.

Günümüzde güvenli iletişimin sağlanmasındaki en önemli kavram kriptografidir. Kriptografi; bilginin güvenliğini sağlamak amacıyla şifreleme ve şifreyi çözme işlemleri ile ilgilenmektedir. Bu nedenle siber uzaydaki güvenlik genellikle kriptografik algoritmalar kullanılarak sağlanır. Modern kriptografinin iletişim güvenliğini sağlamada ortaya koyduğu ana hizmetleri; gizlilik, bütünlük, kimlik doğrulama ve inkâr edememezliktir. Rastgele sayılar ise modern kriptografide, kriptografik algoritmaların şifreleme/şifre çözme anahtarının oluşturulmasında kullanılırlar. Oluşturulan bu rastgele sayılar yazılımsal ve donanımsal olarak üretilebilmektedir. İletişimin kritiklik seviyesine göre üretilen sayıların bit uzunluğu değişkenlik göstermektedir.

Bu tez çalışmasında, herkese açık güvensiz bir hat boyunca iletişimin güvenli bir şekilde sağlanmasını garanti etmek için algılayıcı sensörler kullanılarak kaos tabanlı bir gürültü kaynağı olan Tesla küresinden donanımsal olarak rastgele sayıların elde

edilebileceği amaçlanmaktadır. Elde edilen bu sayıların anlık olarak izlenebilmesi maksadıyla mobil uygulama geliştirilmiştir.

1.2. Tezin Organizasyonu

Bu tez çalışması hiyerarşik bir yapı olarak beş ana bölümden meydana gelmektedir. Birinci bölümde tezin temeli hakkında bilgi verilmektedir. Günümüze dek teknolojik kavramların hızla geliştiği ve gelişen bu teknolojiyle birlikte kritik sistemlerin, gelebilecek saldırı ve tehditlere karşı güvenliğinin alınmasının gerektiği açıklanmaktadır.

İkinci bölümde, kriptoloji kavramının tarihsel gelişiminden bahsedilmiştir. Ayrıca kriptografi ve kriptanaliz kavramlarıyla ilgili teorik bilgiler açıklanmış, rastgele sayı üreteçlerine gereksinim duyan şifreleme algoritmaları sınıflandırılarak detaylı bilgilere yer verilmiştir. Ayrıca tez konusuyla ilgili literatür çalışmalarına da yer verilmiştir.

Üçüncü bölümde, rastgele sayı üreteçlerinin genel sınıflandırılması yapılıp ayrı alt bölümler halinde detaylı olarak açıklanmıştır. Proje kapsamında gerçekleştirilecek olan rastgelelik testleri hakkında teorik bilgiler verilmiştir. Tezde kullanılacak olarak malzeme ve yöntemler hakkında detaylı açıklamalarda bulunulmuştur. Ayrıca projede kullanılan işletim sistemi, programlama dili, kütüphaneler, veritabanı sistemi ve mobil uygulama geliştirme ortamı gibi yazılımsal bileşenlere de değinilmiştir.

Dördüncü bölümde, kullanılan dört farklı kriptografik yöntem ile sensörlerden verilerin elde edilmesi açıklanmış ayrıca bu verilerin rastgelelik testlerin analiz sonuçları hakkında detaylı bilgilere yer verilmiştir.

Son olarak beşinci bölümde ise, genel değerlendirme yapılarak tez çalışması sonlandırılmıştır. Ayrıca gelecek çalışmalara yönelik önerilerde de bulunulmuştur.

2. GENEL BİLGİLER

Bu bölümde kriptoloji kavramının geçmişi, ilk zamanlarda kullanılan şifreleme yöntemleri ve kriptolojinin genel yapısından bahsedilmiştir.

2.1. Kriptolojinin Tarihsel Gelişimi

Günümüzde bilgi alışverişinin daha çok çevrimiçi ortamlarda olması, haberleşmenin güvenliği ihtiyacını daha fazla ortaya çıkarmaktadır. Eski çağlardan beri sağlanan haberleşme gizliliği, günümüzde kişi ve kuruluşlar tarafından daha fazla kullanılmakta ve kullanılmaya da devam etmektedir (Neha, 2017). Bilginin korunması maksadıyla son derece kritik bir hale gelen ve tarihin seyrini önemli etkide değiştiren şifreleme kavramı, yıllar öncesinden başlamıştır ancak kriptografi adı altında Thomas Brownie tarafından 1658 yılında (Sharma ve Garg, 2017) ortaya atılmıştır. İnternet kavramı ortaya atılmadan çok öncesinde değerli bilgileri korumak için ordu tarafından kullanılan kriptografi, günümüzde birçok alanda kullanılmaktadır. Şifre bilimi olarak isimlendirilen kriptoloji, haberleşen iki veya daha fazla nokta arasındaki gizli bilgiyi çeşitli yöntemlerle şifreleyip anlaşılmayacak hale getirerek, bilgi alışverişinin güvenli olarak yapmasını sağlayan, daha sonra alıcı tarafta şifreli metnin çözülmesini sağlayan bilim dalıdır. Temeli yüksek seviyeli matematiksel yöntemlere dayalı tekniklerin ve uygulamaların bütünüdür.

Nesiller boyunca neredeyse her zaman siyasi veya askeri konular, sırlar içermekteydi. Generaller, mesajların düşman tarafından ele geçirilmesini ve anlaşılmasını önlemek amacıyla kendi birliğinin hareketlerini koordine etmek zorundaydı. Kriptografiyi kendi gelişimleri için kullanan meraklı bireyler olsa da askeri ve siyasi kullanımlar dışında, bilgisayar çağından önce kriptografiye gerçekten ihtiyaç duyulmuyordu. Siyasi ve askeri haberleşmeler hala modern şifreleme kapsamındadır, ancak kritik olan bu alanların dışında daha sıradan alanlarda da uygulanmıştır. Kriptografi, elektronik bankacılık ve alışveriş gibi sık sık kullanılan alanlar sayesinde, çoğu insanın farkında olmasalar dahi günlük yaşamına girmiştir. Birçok kullanıcı bilgisayarlarındaki sabit diskleri ya da dosyaları şifrelemekte, bazı insanlar ise gönderdikleri elektronik postaları şifreliyor. Kriptografi günümüzde birçok alanda olmasına rağmen çoğu kriptografi kullanıcısı kriptografinin nasıl işlediği hakkında çok az bilgiye sahiptir. Kriptografinin daha iyi anlaşılması için siber güvenlik uzmanları, ağ yöneticileri ve siber güvenlik gibi daha nitelikli personel gereklidir. Bu profesyoneller

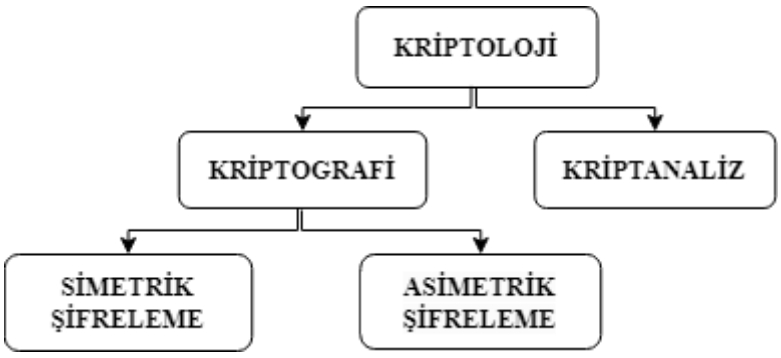
için kriptografi hakkında daha fazla bilgi sahibi olmak en azından kriptografinin uygulanmasına ilişkin daha iyi kararlar almalarını sağlayacaktır. Hassas bilgilerin şifrelenmesi için hangi algoritmanın kullanılacağına ya da hangi anahtar değişim algoritmasının ortadaki adam saldırılarına karşı en dirençli olduğuna karar vermek, kriptografi hakkında biraz daha fazla bilgi sahibi olmayı gerektirmektedir.

Yeni fikirler ve teknik gelişmeler sonucunda kriptolojinin tarihi bölümlere ayrılmıştır:

- Birinci Dünya Savaşı sonuna kadar olan kâğıt ve kalem dönemi,
- Birinci Dünya Savaşı sonundan 1970 yılına kadar elektrik ve mekanik şifreli makinelerin kullanım dönemi,
- 1970 yılından sonraki elektronik dönemdir.

Birinci dönemde adından da anlaşılacağı gibi kâğıtta bulunan yazıyı gizlemek için bazı yöntemler kullanıldı. İkinci dönemde, Alman Enigma elektrik ve mekanik makinesi kullanıldı. Bu elektrik ve mekanik makinelerdeki artış, istatistiksel tekniklerin gelişerek kripto analitik yöntemlerin oluşmasını sağladı. Böylece kriptanaliz, ilk şifreleme cihazının geliştirilmesinde ve yapımında itici bir güç olmuştur. Üçüncü dönem olan elektronik çağda ise veri işleme kavramı ortaya çıkmıştır. Bu çağın başlarında geliştirilen yöntemler halen kullanılmakta veya yenilenerek geliştirilmeye devam edilmektedir. Kriptografi ile matematik arasında güçlü bir bağlantı vardır ve yüksek seviyede artış hızına sahiptir (Diem, 2016).

Kriptoloji bilimi; Şekil 2.1'de görüldüğü gibi kriptografi ve kriptanaliz olmak üzere ikiye ayrılır.



Şekil 2.1. Kriptolojinin sınıflandırılması

Her ne kadar kriptoloji ve kriptografi kelimeleri birçok alanda eş anlamlı olarak kullanılsa da aynı şey değildir. Kriptografi dünyasında kullanılan geleneksel kelimeler aşağıda açıklanmıştır:

Şifrelenmiş metin; açık mesajın, kodlanmış veya şifrelenmiş mesaja dönüştürülmüş halidir.

Kriptanaliz; şifreli metnin anahtar olmadan çözülmesine yönelik fikir ve yöntemlerin incelenmesine denir.

Kriptografi; algoritma ve anahtar olmadan mesajları okunamaz hale getirmek için mesaj değişikliklerine verilen isimdir.

Kriptoloji; hem kriptografi hem de kriptanalizin daha geniş bir alanıdır. Kriptografi ve kriptoloji terimlerini birbirinin yerine kullanmak doğru değildir.

Şifreleme; düz metni şifreli metne dönüştürme işlemidir.

Şifre çözme; şifre çözme ve deşifre etme eşanlamlı terimlerdir. Her iki durumda da şifrelemiş metin düz metne dönüştürülmektedir.

Anahtar; mesajı şifrelemek veya çözmek için algoritmayla birlikte kullanılır. Bir otomobili çalıştıran benzin olarak düşünülürse, bir algoritmayı çalıştıranında anahtar olduğu söylenebilir.

Düz metin; gizli tutmak istenilen verileri içeren orijinal mesajdır.

Bu tanımlamalar hem kriptoloji hem de kriptografi çalışmalarında yer alan en temel kavramlardan bazılarıdır. Kriptolojinin anlanabilmesi için bu temel terimlerin kavranılması gerekmektedir.

2.1.1. Yer Değiştirme Şifreleri

Aynı zamanda ikame şifreleri olarak da bilinmektedir. Bu şifreleme tekniğinde, sıradan metnin her harfini şifreli metinden bir harfle değiştirmek için bir algoritma kullanılır (Yadav ve Ojha, 2013). Tekli alfabe (mono alfabe) ve çoklu alfabe (poli alfabe) ikame şifrelemenin iki farklı türüdür. Tek alfabeli bir ikame şifresinde, düz metindeki belirli bir harf her zaman şifre metnindeki eşleşen harfle değiştirilir. Örneğin normal metindeki bir ‘a’ harfi şifreleme metninde her zaman bir ‘k’ harfi olacaktır. Çok alfabeli ikamede çeşitli yöntemler kullanılır, örneğin düz metindeki bir ‘a’ harfi şifreli metinde bazen ‘k’ bazen de ‘j’ olabilir.

2.1.2. Sezar Şifreleme

Sezar şifreleme en iyi bilinen eski şifreleme teknikleri arasındadır (Gowda, 2016). Julius Caesar bu şifreleme yöntemi ile askeri mesajları, düz metnin tüm karakterlerini üç boşluk sağa kaydırarak şifrelemiştir. Örneğin sezar şifreleme yöntemiyle;

'Deneme bir iki'

'Ghqhph elu lnl'

olarak şifreli hale dönüşecektir. Düz metindeki 'k' harfinin üç harf sağa kaydırılarak 'n' haline getirildiği görülmektedir. Daha sonra, normal metindeki 'o' harfi üç harf sağa kaydırılarak şifreli metindeki 'r' haline getirilir. Bu durumda kaydırmaların hiçbiri 'z' harfini geçmemiştir. Eğer giriş metninde 'y' harfi üç pozisyon sağa kaydırılırsa sonuç ne olacaktı? İşlem a harfinden başlayarak tüm alfabeyi dolaşır, sonuç olarak şifreli metinde y harfi b harfine dönüşecektir. Sezar yönteminde her ne kadar üç boşluk sağa kaydırılarak şifreleme gerçeklense de herhangi sayıda sağa veya sola kaydırma tekniğiyle bu işlem gerçekleştirilebilmektedir. Kriptoloji konusuna başlamak için sezar şifreleme kolay anlaşılır bir yöntem olduğu için iyi bir seçenek olacaktır. Sezar şifresi, günümüz kriptografik isterleri için uygun olmamasına rağmen, bir kriptografi algoritması için gerekli tüm temel hususları içerir. Verilen örnekte, düz metin mesajı 'deneme bir iki' dir. Algoritma verilen bu düz metnindeki her harfi 3 adım sağa kaydırarak 'ghqhph elu lnl' olan şifreli metni elde etmektedir. Mevcut olan bütün simetrik algoritmalar kabaca bu şekilde bir yapı kullanır. Sezar şifresi ile çağdaş simetrik şifreler arasındaki tek fark, algoritmanın karmaşıklığı ve anahtarın boyutudur.

2.1.3. Atbash Şifreleme

M.Ö. 500 ile 600 yılları arasında İbraniler tarafından (Morkel ve Eloff, 2004) ortaya çıkarılan bu şifreleme yönteminde şifreleme için diziyi alfabenin ters yönünde çevirmek yeterlidir. Günümüz şartlarına göre bu yöntem sezar şifreleme gibi kırılması oldukça kolay bir yöntemdir. Örneğin 'a' ve 'b' harflerinin İngilizce alfabesindeki karşılıkları 'z' ve 'y' harfleridir. Örneğin atbash şifreleme yöntemiyle;

'Deneme bir iki'

'Wvmvnv yri rpr

olarak şifreli hale dönüşecektir. Görüldüğü gibi alfabedeki beşinci harf olan 'e' harfi 'v' ile, alfabedeki ikinci harf olan 'b' harfi ise 'y' ile değiştirilerek işlemler tekrarlanmaya devam eder. Atbash şifrelemede, günümüz kriptografik isterleri için uygun olmamasına

rağmen, bir kriptografi algoritması için gerekli tüm temel hususları içerir. Çağdaş simetrik şifreler ile arasındaki fark, algoritmanın karmaşıklığı ve anahtarın boyutudur.

2.1.4. Affine Şifreleme

Affine şifreleme alfabedeki her harfin bir sayısal değere eşlenerek matematiksel bir işleve uğradığı ve ardından tekrar bir harfe dönüştürüldüğü bir şifreleme yöntemidir (Mezaal ve Abdulkareem, 2017) Affine şifrelemedeki temel formül; ax + b (mod m) kullanılmaktadır. 'M' alfabenin uzunluğunu, 'x' düz metindeki harfin sayısal karşılığını, 'b' kaydırılması gereken miktarı, 'a' ise kaç kat olması gerektiğini ifade etmektedir (Arroyo ve Delima, 2020). Seçilecek olan 'a', 'm' ile ortak çarpanları olmayan bir asal sayı olmalıdır. Örneğin 12 sayısında 3 ve 4 bileşenleri mevcuttur. 7 sayısı seçilecek olursa 12 sayısı ile asaldır ve ortak bileşenleri yoktur. Affine şifreleme de ikame şifrelemenin bir türü olduğundan, bu sınıfın zayıflıklarını miras almaktadır.

2.1.5. ROT 13 Şifreleme

Basit bir ikame şifreleme tekniği olan ROT 13 her harfi 13 derece sağa döndürerek çalışmaktadır (Mantoro ve Lazuardi, 2018). Şifrelemedeki temel formül; 1x + 13 (mod 26) kullanılmaktadır. Örneğin ROT 13 şifreleme yöntemiyle;

'Deneme bir iki'

'Qrarzr ove vxv'

olarak şifreli hale dönüşecektir. 26 harften oluşan bir alfabe kullanıldığı zaman ROT 13'ün iki kere uygulanması sonucu, şifrelenmiş olan mesaj çözülebilir. Bu yüzden günümüz standartlarına göre pek kullanışlı değildir. ROT 13'ün Microsoft Windows Kayıt Defterindeki bilgilerin bir kısmını şifrelemek ve Netscape Communicator tarafından elektronik posta şifrelerini kaydetmek için kullanıldığı belirtilmektedir. Sezar şifreleme yönteminin özel bir türüdür.

Birçok insan uzun yıllar boyunca sırlarını korumaya çalışmıştır. Birinci Dünya Savaşının sona ermesi ve makinelerin ortaya çıkışıyla yeni bir mesajlaşma yeteneği ortaya çıkmıştır. İkinci Dünya Savaşında Almanya tarafından gizli mesajlaşma sistemi olarak kullanılan bir şifreleme ve şifre çözme özelliğine sahip Enigma makinası, savaş boyunca askeri amaçla kullanıldı. Enigma, Latince gizemli anlamına gelen 'aenigima' kelimesinden türetilmiştir. Birinci Dünya Savaşından hemen sonra şirketler ve bankalar arasındaki ticari haberleşmenin güvenliğini sağlamak maksadıyla ilk olarak Almanya'da geliştirilen mekanik rotorlu bir şifreleme mekanizmasıdır. Alman ordusu 1926 yılında

kendine özgü olması ve yüksek güvenlik gücünün bulunması sebebiyle Enigma makinesine ilgi duymaya başladı (Prasad ve Kumari, 2020). Enigma ile şifrelenen bir mesajın orijinal halini bulmak için uzun süren denemeler yapılması gerekiyordu. Bu özelliğinden dolayı Alman istihbaratı Enigma sistemiyle haberleşmenin güven altında olduğunu düşünüyordu. Hatta Fransız ve İngiliz kriptanalistler Enigma şifrelerinin kırılamaz olduğuna öylesine inanmışlardı ki bu konudaki çalışmalarını sonlandırmışlardı. 1939 yılında Enigmayı kırabilmek için İngilizler tarafından kurulan merkezin en önemli ismi Alan Turing olmuştur. İkinci Dünya Savaşı sırasında Alman şifrelerinin kırılmasında çok önemli bir rol oynayan Alan Turing, Cambridge matematikçisi, bilgisayar bilimcisi ve kriptologdur (Grime, 2013).

2.2. Kriptografi ve Kriptanaliz

"Kriptografi" kelimesi gizli yazı anlamına gelen, "secret" ve "writing" kelimelerinden türetilmiştir. Bir gönderici açık ağlar üzerinden bir alıcıya ileti göndermek istediği zaman, dinlenme ve değiştirilme tehdidi altında kalır. Burada tehdit altında kalacak söz konusu ileti düz metindir. Bazı tanımlamalarda düz metne plaintext adı da verilmektedir. Bir iletinin içeriğini saklamak üzere yapılan gizleme işlemine de şifreleme (encryption) denir. Bu işlem düz metni başkalarının anlamayacağı şifreli hale dönüştürmektedir. Bu bilgi bir yere iletilmek amacıyla şifrelenen bir mesaj veya saklanmak amacıyla şifrelenen bir veri olabilir. Alıcı tarafın, şifreli metni düz metne geri çevirme işlemine ise şifre çözme (decryption) denir (Atar vd., 2017). Şifre kırmak, mesajı çözmek maksadıyla kaba kuvvet saldırılarından daha etkili olan herhangi bir yöntem bulmak anlamına gelmektedir. Eğer kullanılan algoritmanın anahtar uzunluğu 128 bit ise, 2^{128} olası anahtar var demektir. Bu olasılık ile anahtarın kırılabilmesi için çok uzun yıllar gerekir. Kriptanaliz olarak adlandırılan şifreli metnin çözümünü, çok büyük matematiksel işlemlere dayanır, temel amaç kriptografik sistemlerdeki zayıflıkları ve ortaya çıkabilecek bilgi kaçaklarını elde etmektir. Yeni tasarlanan bir algoritmanın gücünü test etmek maksadıyla bu algoritmalar kriptanaliz tekniklere tabi tutulmaktadır.

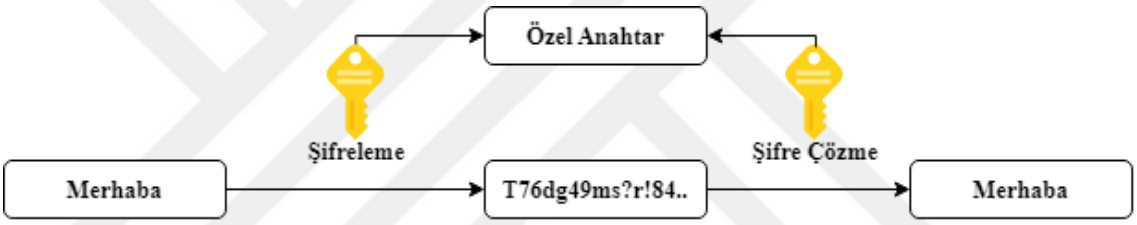
2.3. Şifreleme Algoritmaları

Son zamanlarda ortaya çıkan Nesnelerin İnterneti ve Yapay Zekâ Uygulamalarıyla birlikte şifreleme algoritmalarının önemi gittikçe artmaktadır. Geliştirilen bu uygulamalarda cihazlar/sensörler genellikle dağıtık yapıda olduğundan, bu uygulamalara saldırılar artmaya başlamış, oluşabilecek güvenlik tehditlerini ortadan kaldırmak

maksadıyla arayışlara gidilmiştir. Güvenlik tehditlerini bertaraf etmek için bu zamana kadar birçok şifreleme sistemi geliştirilmiştir. Geliştirilen bu sistemlerin birçoğu eskiyip güvenliğini yitirmiş, bazıları ise saldırılara karşı dayanıklı ve kırılamamış algoritmalardır. Şifreleme algoritmaları; tek anahtar kullanan simetrik algoritmalar ve iki fakat birbiriyle bağlantılı anahtar kullanan asimetrik algoritmalar olmak üzere iki kategoriye ayrılır.

2.3.1. Simetrik Şifreleme Algoritmaları

Simetrik şifreleme algoritmalarında, iletilmek istenen şifreli mesaj şifreleme algoritması tarafından bir dizi işleme tabi tutulur. Bu işlemler sırasında mesaj, Şekil 2.2'de gösterildiği gibi gönderici tarafından şifreleme anahtarıyla şifrelenir. Alıcı, kendisine ulaşan şifreli mesajı orijinal haline döndürürken kendisinde bulunan aynı şifreleme anahtarıyla mesajı çözer. Yani simetrik anahtarlı şifreleme algoritmalarında şifreleme ve çözme işlemlerinde aynı anahtarlar kullanılır (Yılmaz ve Ballı, 2016).



Şekil 2.2. Simetrik şifreleme ve şifre çözme

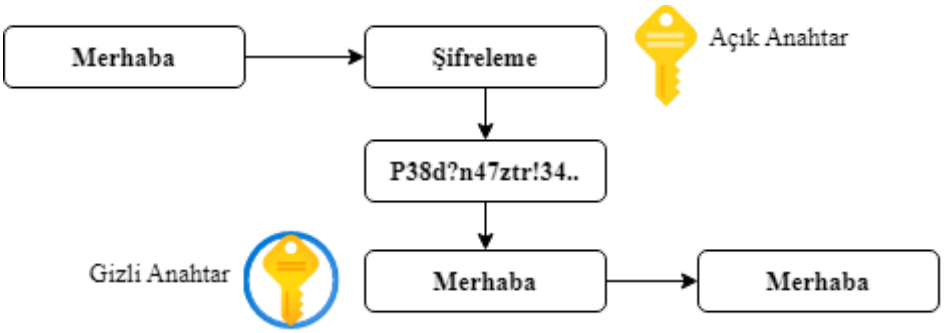
Günümüzde ismi duyulmuş simetrik şifreleme algoritmaları; AES (Advanced Encryption Standard), DES (Data Encryption Standard) ve 3DES (Triple DES)'tir. DES, 1970 yılında IBM'de bulunan mühendisler tarafından geliştirilmiş, 1977 yılında ise NIST (National Institute of Standards and Technology) tarafından kabul edilmiştir. Güvenlik ihtiyaçlarının fazla olduğu finansal hizmetlerde ticari olarak kullanılan ilk şifreleme algoritmasıdır. Bir blok şifreleme algoritması olan DES, 56 bit anahtar uzunluğuna sahiptir. Blok şifreleme teorisinin temellerini atmıştır, bu teorinin geliştirilmesine de katkıda bulunmuştur. Anahtar boyutunun küçük olması, algoritmayı kaba kuvvet saldırılarına karşı savunmasız hale getirmiş, bilgi işlem gücünün hızlı bir şekilde artmasıyla da birçok saldırgan tarafından saldırı yapılmasına sebep olmuştur. Bu sebeple hassas ve gizli verilerin güvenliği ve gizliliğini sağlamak için kullanılamaz olduğu sonucuna varılmıştır (Sharma ve Garg, 2017). Bu olumsuzluklardan dolayı DES çok kırılgan bir hale gelmiş, zaman geçtikçe 3DES ve AES algoritmalarıyla değiştirilmeye başlanmıştır.

3DES, 1978 yılında yine IBM tarafından geliştirilmiş olup 3 adımda (şifrele-çöz-şifrele mantığı) çalıştığı için bu isimle adlandırılmıştır. DES algoritmasının biraz daha gelişmiş sürümü olarak ortaya çıkmıştır. 3DES algoritmasında tüm bloklara 3 ayrı kere DES uygulanır. DES algoritmasına göre anahtar boyutunun daha uzun olması, gelebilecek saldırılara karşı daha dayanıklı olduğunun kanıtıdır. Söz konusu üç anahtar birbirinden bağımsızdır ve toplamda 168 boyuta sahiptir.

AES ise Amerika Birleşik Devletleri tarafından gizli olarak nitelendirilen bilgi ve belgelerin güvenliğini sağlamak için 2001 yılında ortaya çıkmıştır. İki Belçikalı kriptograf olan Joan Daemen ve Vincent Rijmen tarafından geliştirilmiştir. Şifreleme uzmanı olan Joan Daemen blok şifreleme, akış şifreleme ve özetleme fonksiyonları üzerine çalışmalar yapmıştır. Günümüzde en çok kullanılan simetrik şifreleme yöntemi olan ve aynı zamanda Rijndael blok şifreleme olarak da bilinen AES, güvenlik ve hız açısından yüksek verimliliğe sahiptir ve halen kırılamamıştır. 2001 yılında NIST tarafından DES'in yerini alması için önerilmiştir. Şifreleme ve şifre çözme sırasında 128, 192 ve 256 bit olarak üç farklı anahtar boyutu kullanılabilir (Abood vd., 2018).

2.3.2. Asimetrik Şifreleme Algoritmaları

Asimetrik şifreleme algoritmaları, aynı zamanda açık anahtarlı şifreleme olarak da adlandırılmaktadır. Bu şifreleme türünde iki farklı anahtara dayalı şifreleme sistemi kullanılmaktadır. Bu sistemde şifreleme için bir tane genel anahtar ve bundan farklı olarak şifre çözmek için de bir tane özel anahtar bulunmaktadır. Şifreleme anahtarı herkese açık olmalıdır, bu yüzden açık anahtarlı şifreleme adını almıştır. Özel anahtar ise sadece şifre çözme işlemini yapacak kişi tarafından kullanılan ve güvenli bir şekilde saklanması gereken anahtardır. Asimetrik şifreleme algoritmalarında bilgisayarın işlem gücünü artıran çok büyük asal sayılar kullanılmaktadır (Atar vd., 2017). Anahtar boyutunun uzun olması ve hesaplama sürelerinin fazla olması nedeniyle asimetrik kriptografide açık anahtar altyapısı kullanılır (Maqsood vd., 2017). Şekil 2.3'de asimetrik şifrelemenin temel yapısı gösterilmiştir.



Şekil 2.3. Asimetrik şifreleme ve şifre çözme

RSA, açık anahtarlı şifreleme olarak da bilinen asimetrik şifrelemede en yaygın algoritmadır. Tanınmış ve saygın kriptograflar olan Leonard Adleman, Adi Shamir ve Ron Rivest tarafından 1977 yılında ilk kez yayımlanmıştır. Rc2, Rc4, Rc5, Md2, Md4, Md5 ve Md6 MIT profesörü olan Ron Rivest'in tasarladığı algoritmalardan yalnızca birkaçıdır. İsrailli kriptograf Adi Shamir diferansiyel kriptanalizi geliştirmesiyle tanınır. DNA'nın bir bilgi işlem sistemi olarak kullanımında önemli gelişmeler Leonard Adleman tarafından gerçekleştirilmiştir. RSA algoritması, çok büyük asal sayılarla, yüksek hesaplama gücü gerektiren çarpanlara ayırma ve üssel işlemlere dayanır.

Bir şifreleme algoritmasında bulunması gereken temel özelliklerden bazıları aşağıda tanımlanmıştır:

- Gizlilik: Bilgi, sadece izin verilen kişiler tarafından okunabilmelidir.
- Bütünlük: Bilgi, iletimi sırasında yetkili kişiler dışında başka kimse tarafından değiştirilmemelidir.
- Reddedilemezlik: Bilgiyi üreten ya da ileten kişinin daha sonrasında bunu inkâr edememesidir.
- Erişilebilirlik: Yetkili kişilerin ihtiyaç duyduğu anda gerekli bilgiye ulaşabilmesidir.
- Kimlik denetimi: Gönderici ve alıcı taraf birbirlerinin kimliklerini doğrulaması işlemidir (Coşkun ve Ülker, 2013).

2.4. Literatür Taraması

Rehman ve arkadaşları yaptıkları çalışmada; Ardunio/Raspberry Pi cihazına bağlanan bazı sensörler yardımıyla sıcaklık, toprağın nem değeri ve suyun pH değerini sahadan toplayıp GSM ortamı ile sunuculara gönderilmesini sağlayan bir sistem tasarlamışlardır. Sunuculara gelen bu veriler web üzerinden grafiksel olarak toprak

sahiplerine önem arz edecek bilgiler vermektedir. Böylece arazi sahibi ek zaman ve emek harcamadan elde ettiği parametreler ile hasadı arttıracak önlemler alabilecektir (Rehman vd., 2020). Bu çalışmada nem sensörü le 0-1.023 arası, pH sensörü ile 0-14 arası ondalık ve sıcaklık sensörü ile 0°C ile 50°C arası ondalık elde edilen değerler, bir rastgele sayı üretecinin tohum değeri olarak kullanılabilir.

Sunny ve arkadaşlarının yaptıkları çalışmada; nükleer atık depolama konteynerin çevresel parametreleri, düşük güç ve maliyet ile gerçek zamanlı olarak uzaktan izlenebilmektedir. Tasarlanan bu sistem enerji toplama birimi, pil ve sensörlerden oluşmaktadır. Hidrojen gazını ölçmek için MQ-8 sensörü, basınç, sıcaklık ve nem değerlerini ölçmek için ise BME680 sensörü kullanılmaktadır. Sensörler vasıtasıyla alınan veriler, kablosuz iletişim sensörü ile merkeze iletilip uzaktan izlenebilmektedir (Sunny vd., 2020). Bu çalışmada sensörlerde; hidrojen değeri için 100-10.000 ppm, sıcaklık değeri için -40°C ile 85°C, nem değeri için %0-100 rH, basınç değeri için 300-1.100 hPa arası elde edilen değerler bir rastgele sayı üretecinin tohum değeri olarak kullanılabilir. BME680 sensörü için ölçülen; sıcaklık 22,096°C, nem %13,931, basınç 995,267 değerleri örnek olarak verilebilir.

Üçgün ve arkadaşları yaptıkları çalışmada; IoT tabanlı iç ortam havası izleme sistemi tasarlamışlardır. Tasarlanan sistem; Raspberry pi 3, sıcaklık, nem, ışık ve MQ2 gaz sensörlerinden oluşmaktadır. Bu sensörler vasıtasıyla alınan iç ortam değerleri, Raspberry pi cihazıyla veri tabanına işlenir. Php ve html dilleriyle geliştirilen web arayüzü ile anlık olarak iç ortam verileri izlenerek kullanıcılara bilgiler vermektedir (Üçgün vd., 2020). Bu çalışmada sensörlerde; sıcaklık değeri için 0-50°C, nem değeri için %20-90 rH, CO gazı ve duman değerleri için 100-10.000 ppm, ışık değeri için 10-1.000 lux arası elde edilen değerler bir rastgele sayı üretecinin tohum değeri olarak kullanılabilir.

Tuncer ve Genç yaptıkları çalışmada; insan hareketleri tabanlı bir rastgele sayı üreteci önermiştir. Önerilen üreteç, dünyada birçok insanın kullandığı mobil telefonlardaki GPS ve ivme sensörünü kullanmaktadır. İlk aşamada, mobil telefonu taşıyan kişinin hareketleri sonucunda ivme ve konum değişimleri android tabanlı bir mobil cihazdan örneklenerek sayı dizileri elde edilmiştir. Son olarak, sayı dizilerinin istatistiksel özelliklerini iyileştirmek için XOR son işlemi uygulanmış ve rasgele sayı üretimi gerçekleştirilmiştir. Önerilen gerçek rastgele sayı üreteci, cep telefonu platformu için uygun, evrensel ve düşük maliyetli olup kişiye özgü rasgele sayı üretmektedir. Sayıların kalitesini ölçmek amacıyla yapılan testlerde; kişinin koşma ve yürümesi

esnasında elde edilen sonuçlar başarılı olmuş ancak durma anında çok küçük hareketler GPS den elde edilen konumlarda herhangi bir değişiklik göstermediğinden başarısız olmuştur (Genç ve Arslan Tuncer, 2019).

Yaşar ve arkadaşları yaptıkları çalışmada; C programlama dilini kullanarak random fonksiyonu ile rassal sayılar elde etmiştir. Bu rassal sayılar elde edilirken Middle-Square (orta kare) metodundan faydalanılmıştır. Elde edilen rastgele bit dizileri Sha256 hash fonksiyonu ile sabit çıkışlı veri haline dönüştürülmüştür. Elde edilen çıktılardan 5 adedi Ki-Kare testi ile analiz edilip güvenilirliği hesaplanmıştır. Sonuç olarak hesaplanan verilere göre iyi sonuçlar elde edilmiş ve Sha256 algoritmasının güvenilirliği kanıtlanmıştır (Yaşar vd., 2021).

Chen yaptığı çalışmada; yüksek çözünürlüklü bir webcamden video ve ses gürültüsü elde etmiştir. Elde edilen analog sinyaller rastgele sayı üretiminde kullanılmak üzere sayısal hale dönüştürülmüştür. Görüntü verisi için, 320*200 piksel boyutunda saniyede ortalama 8 ile 15 arası kare alınır ve alınan kareler üzerindeki belirli pikseller üzerinde yapılan matematiksel işlemler sonucu tohum değeri elde edilmiş olur. Ses verisi için ise alınan her video dosyasında saniyede 30 farklı ses değeri üretilir. Kaynaktan alınan bu ses verileri 16 bitlik (2 bayt) stereo formatına çevrilir ve tohum değeri elde edilmiş olur. Bu çalışmada gerçek rastgele sayı üretimi için, karmaşık ekipmanlar yerine üzerinde mikrofonu ve webcamı olan bir kişisel bilgisayar yeterlidir. İleriki süreçlerde, aynı ekipmanlara sahip tablet ve akılı telefonlar üzerinde de aynı uygulama gerçekleşebilir (Chen, 2013).

Etem ve Kaya yaptıkları çalışmada; rastgele sayı üretecini bir donanım kullanmadan LCG (Linear congruential generator-Doğrusal uyumlu üreteç) algoritmasıyla oluşturduktan sonra Trivium algoritmasıyla son işlemden geçirerek tasarlamışlardır. Rastgele sayı üretme amacıyla kullanılan ve oldukça temel matematiksel işlemlere dayanan LCG algoritması, sistem kaynaklarını minimum seviyede kullanmasından dolayı çok avantajlıdır. Ancak temel matematiksel işlemlere dayandığı için kritik kriptografik sistemlerde kullanılması uygun değildir. Bu zafiyetlerin giderilmesi için Trivium algoritması kullanılarak istatistiksel özellikler iyileştirilmiş ve kritik sistemlerde kullanılabilir seviyeye getirilmiştir (Etem ve Kaya, 2020).

Ansari ve arkadaşları yaptıkları çalışmada; Ardunio mini bilgisayarına bağlı düşük maliyetli sensörleri kullanarak gerçek rastgele sayı üreteci tasarımı yapmışlardır. Dört adet ldr sensörü ve denetleyicinin pinlerine bağlı ses sensörü tasarımda kullanılan donanımlardır. Işık (ldr) ve ses sensörleriyle doğada bulunan gürültüler alınarak elde

edilen sinyaller ile rastgele bitler üretmek için son işlem algoritmasına aktarılmıştır. Söz konusu ses sensörü insan kulağının duyma seviyesinde çalışacağından yakın ortamdaki analog ses yoğunluğunu sayısal veriye dönüştürür. Modüler aritmetik ve XOR yöntemlerinin kullanımıyla deterministik olmayan sayı elde edinimi sağlanmıştır (Ansari vd., 2022).

Wibowo ve arkadaşları yaptıkları çalışmada; Android akıllı telefonunda bulunan ses girişinden sözde rastgele sayıların oluşturulması hakkında çalışma yapmıştır. Ses girişi olan mikrofon kullanılarak rastgele gürültü elde edilmiş, elde edilen bu analog değerler bit değerine dönüştürülmüştür. Analog seslerin dijital hale getirilmesini ADC (Analog to Digital Converter) isimli araç sağlamaktadır. Ses girişinden elde edilen bu gürültüler sayısal hale getirildikten sonra son işleme tabi tutulmuştur. Son işlem olarak 1998 yılında Matsumoto ve Takuji Nishimura tarafından bulunan Mersenne Twister algoritması kullanılmıştır. Bu algoritma ile tohumun başlangıç seviyesi ortadan kaldırılarak rastgele dizilerin üretilmesi sağlanmıştır (Wibowo ve Romadhon, 2018).

Zhang ve arkadaşları yaptıkları çalışmada; akıllı telefonlarda bulunan kameraya dayalı bir taşınabilir rastgele sayı üreteci önermişlerdir. Operatör olarak adlandırılan kişi baş parmağını hem görüntü sensörüne hem de arka kameranın yanında bulunan flaş lambasına bastırması sonucu gürültü olayı başlatılmış olur. Lambadan çıkan ışık baş parmak tarafından oldukça zayıflatıldıktan sonra küçük bir kısmı ise görüntü sensörünün yakalayacağı şekilde geri yansıyacaktır. Bu sayede periyodik şekilde oluşan görüntünün piksel olarak renk karşılıkları olacaktır (Zhang vd., 2014).

Katyal ve arkadaşları yaptıkları çalışmada; içerisinde 20'den fazla balık bulunan bir akvaryumu ortam kaynağı olarak kullanarak tohum verilerini elde etmek için görüntüler yakalamıştır. Rastgele sayı dizileri oluşturmak maksadıyla her seferinde farklı bir görüntü verisi elde edilmiştir. Elde edilen verilerden oluşan gerçek rastgele tohum daha sonra, bir rastgele sayı dizisi vermesi maksadıyla bir algoritmayı besleyerek 16 bitlik çıkışlar alınması sağlanmıştır. Örnek olarak 640x480 piksele sahip çözünürlüklerde 16338 ve 3453 tohumları alınmıştır (Katyal vd., 2013).

Park ve arkadaşları yaptıkları çalışmada; CMOS görüntü sensörünün karanlık gürültüsünü kullanarak gerçek bir rasgele sayı üreteci sunmuşlardır. Önerilen TRNG, görüntü sensörünün karanlık özelliklerine dayandığından, gerçek rastgeleliği sağlamak için ışık ve optik kaynağı gibi herhangi bir ek donanım gerektirmeden mobil telefonlar ile sağlanmaktadır. Elde edilen görüntü verileri bilgisayara aktarılarak her pikselin görüntü verisi analiz edilir. İşlem sonrası oluşan blokta, rastgeleliği artırmak ve

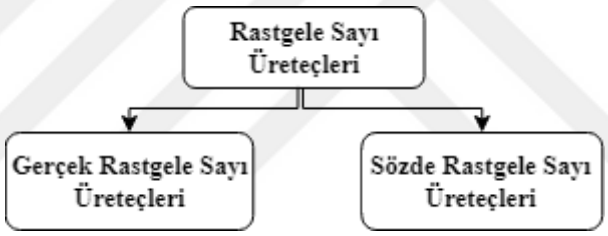
uygulanabilir bir veri formatı oluşturmak için, belirli algoritmalardan geçirilir (Park vd., 2019).

Özkaynak ve arkadaşları yaptıkları çalışmada; mobil cihaz kullanarak rastgele sayı üreten algoritmayı önermiştir. Rastgele sayı üretimi iki aşamada gerçekleştirilmiştir. İlk adımda bir entropi havuzu oluşturulmuş, ardından alınan değerler son işlem için SHA-3 hash fonksiyonuna verilmiştir. Önerilen yaklaşım, mobil cihazların kamera işlevini kullanıp iki ardışık fotoğraf çekerek bir entropi havuzu oluşturmaktadır. İki görüntünün doğası gereği bazı piksellerinin ortama ve cihaza bağlı olarak değişeceği açıktır. Başka bir deyişle, tekrarlaması mümkün olmayacaktır. Önerilen uygulamada entropi havuzu, kullanıcı tarafından çekilen iki fotoğraf arasında farklı olan ilk piksel değeri ile oluşturulmaktadır. Olası saldırıları engellemek için bu veriye zaman bilgisi eklenmesi sayesinde kullanıcının başarılı bir şekilde aynı fotoğrafı çekebilme ihtimali ortadan kaldırılmaktadır. SHA-3 algoritması bu değeri girdi olarak alır ve sonuç olarak 512 bit uzunluğunda rastgele tamsayılar üretir (Ozkaynak vd., 2015).

Sırbistan'ın Similjan köyünde 1856 yılında dünyaya gelen Nikola Tesla, elektrik gerilimini yükselterek düşük akım yoğunluğuna sahip bir çıkış ile elektrik gücünün kablosuz olarak aktarılmasını sağlamıştır (Sezgin, 2021). Bu makalede, entropi kaynağı olarak kullanılacak olan Tesla küresinin merkezinde bulunan küçük bir küreden dış kısımda bulunan cam küreye doğru mavi ve kırmızının tonlarında farklı renklerde meydana gelen rastlantısal elektriksel ışımalar ile ham verilerin elde edilme aşamaları değerlendirilmiştir. Gürültü kaynağı olarak fiziki ortamda bulunan Tesla küresinden IoT cihaz veya sensörler ile ham veriler alınarak elde edilen değerler Raspberry Pi cihazı yardımıyla sayısal verilere dönüştürüldükten sonra ihtiyaç halinde son işlem algoritmasına tabi tutulup sabit uzunlukta sayı dizileri elde edilecektir.

3. MATERYAL VE YÖNTEM

Rastgele Sayı Üreteçleri şans oyunları, bilgisayar oyunları, kriptografi gibi rastgeleliğe ihtiyaç duyulan her yerde kullanılabilir. Kriptografide, algoritmalarla birlikte kullanılarak haberleşmenin gizli bir şekilde yapılmasını amaçlayan, yazılımsal ya da donanımsal olarak üretilen belirli uzunluğa sahip karakter dizileri olarak adlandırılır. Bu rastgele dizileri gerçek olacak şekilde bir bilgisayar programı kullanarak üretmek imkansızdır. Bilgisayar yazılımı ile gerçek bir rastgele sayı üretilemez, bunun için donanım kullanılması mecburidir. Gerçek bir rastgele sayı üretimi için doğada meydana gelen bazı olaylardan veri almak kaçınılmazdır. Kullanım alanına göre rastgele sayıların kalitesi ya da anahtar uzunluğu çok büyük öneme sahip olabilmektedir. Üretilecek olan sayıların rastgele olabilmesi için tahmin edilemez ve birbiriyle herhangi bir bağ olmayacak şekilde üretilmesi beklenir. Rastgele sayılar, Şekil 3.1'de gösterildiği gibi "sözde" veya "gerçek" olmak üzere iki ayrı şekilde üretilmektedir.



Şekil 3.1. Rastgele sayı üreteçlerinin sınıflandırılması

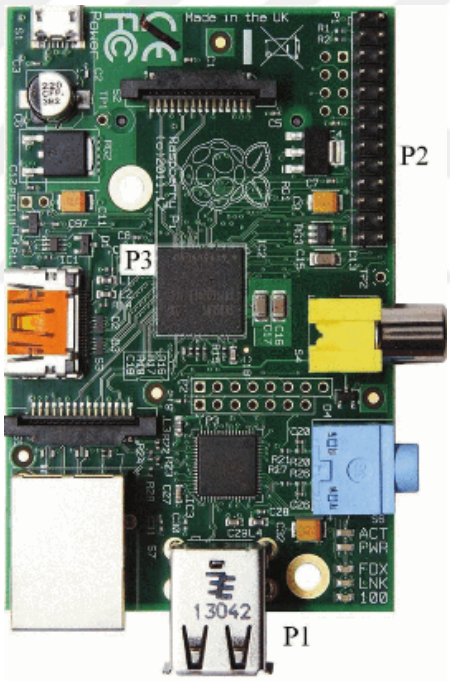
Sözde RSÜ'ler çıktılarını bir algoritma aracılığıyla oluşturur ve bundan dolayı belli bir süre sonra çıkış verisi periyodikleşir. Gerçek RSÜ'lerin temelinde rastgeleliği sağlayan bir belirsizlik kaynağı vardır. Bu belirsizlik veya entropi kaynakları genel olarak fiziki dünyadaki gürültü davranışlarıdır. Gerçek RSÜ'de rastgelelik kaynağı, kaotik bir belirsizlik kaynağına dayandığından, çıkış verisinin periyodik olmaması beklenir (Tavas, 2011). Üretilen rastgele sayılar; kamu, askeri ve sivil sektörlerin bilgi güvenliğini hedefleyen haberleşme sistemlerinde şifreleme anahtarı olarak kullanılabilir. Tasarlanan sistemlerin/donanımların güvenilirliği, kriptografik sistemlerin test edilmesinden sorumlu makam tarafından onaylanması sonucu kanıtlanmış olur. Böylece onaylanan sistem veya donanım, kritik olarak görülen kamu ve askeri haberleşme altyapılarının güvenliğini sağlamak amacıyla kullanılabilir.

3.1. Materyal

Bu bölümde Raspberry Pi donanımları ve yazılımsal bileşenleri hakkında bazı genel bilgiler verilmektedir.

3.1.1. Raspberry Pi 4 Model B

Raspberry Pi, 2012 yılında sunulan Şekil 3.2'de gösterildiği şekilde güçlü, küçük ve eğitim odaklı bir bilgisayar kartıdır. Komut girişi için bir ekran birimi, güç kaynağı ve klavye gerektirir. Standart bir bilgisayar ile benzer performansa sahip, kredi kartı büyüklüğünde, birçok performansa sahip bir şekilde cihazlarla birlikte arayüz oluşturabilmektedir. Model A ve Model B bellek çeşitleriyle birlikte ses ve iletişim donanımı, merkezi ve grafik işlem birimleri gibi sistem bileşenlerinin büyük çoğunluğunu tek bir bileşen içerisinde kullanıma sunulmasıyla oluşmuştur (Maksimović vd., 2014).



Şekil 3.2. Raspberry Pi 4 Model B

Raspberry Pi'nin klasik bilgisayarlara göre en büyük avantajı düşük maliyetli olmasıdır. Yukarıdaki şekilde bizim projemizde kullandığımız Raspberry Pi 4 Model B, 4 GB ram kapasitesi ile donatılmıştır. P1 iki USB bağlantı noktası, P2 bir GPIO bağlantı noktası, P3 kısmının alt kısmında ise ARM işlemcisi bulunmaktadır. Daha fazla USB bağlantı noktasına ihtiyaç duyulduğunda USB hub gereklidir (Marot ve Bourennane, 2017).

3.1.2. Tesla Küresi

Elektrik enerjisi kablolu ve kablosuz olmak üzere iki şekilde aktarılmaktadır. Nicola Tesla 1891 yılında tesla bobinini icat ettikten sonra kablosuz güç teknolojisi geçerli hale geldi. Havada dolaşan elektriği göstermek için kullanılan bu icat, floresan ampullere kablosuz olarak gücü aktarabilir. Tesla küresi olarak adlandırılan Şekil 3.3'te ki cihaz düşük basınçlıdır ve 8 Watt elektrik tüketimine sahiptir. Çalıştırıldığında içinde bulunan gaz elektromanyetik olarak uyarıldıktan sonra ortaya çıkan renkli rastlantısal ışımalar oluşur. Kürenin ortasında bulunan merkezi küçük küre elektrot görevi görmektedir. Cihaz çalıştırılmaya başlandığı anda merkezde bulunan küreden dışardaki cam kürenin iç yüzeyine doğru elektrik alan çizgileri doğrultusunda rastlantısal renkli ışımalar şeklinde plazma tanecikleri iletilmeye başlar. Dışta bulunan küreye parmak ile dokundurulduğu zaman merkezdeki küreye doğru tekli ışıma meydana gelir.

Şekil 3.3. Tesla küresi

Bu projede elektriğin kablosuz olarak küre dışına aktarılması sonucu oluşan rastlantısal ışımaların kamera ile anlık değişen hareketli konumları ve oluşan ışımaların ışık sensörü ile ölçüm sonucu elde edilen rastgele değerler ile rastgele sayı üreteci için ham tohum değerinin elde edilmesi araştırılacaktır.

3.1.3. RGB Sensörü

TCS34725 sensörü kırmızı, yeşil ve mavi (RGB) renkleri algılayarak (Panuganti vd., 2022) bu renklerin sayısal değerlerinin sonucunu döndürür. Sensör üzerine entegre

edilmiş olan kızılötesi filtre sayesinde, rengi algılanmak istenen cisimden yansıyan ışınları bloke ederek renk ölçümünün doğru bir şekilde yapılması sağlanır. Şekil 3.4'de ki RGB sensörü üzerinde bulunan VIN, GND, SCL ve SDA pinleri (Qutieshat vd., 2019) Raspberry Pi üzerinde bulunan 3V3, GND, SCL ve SDA pinlerine sırasıyla jumper kablolar yardımıyla bağlanmalıdır. SDA pini veri iletimini sağlarken SCL pini ise saat sinyali işlemini gerçekleştirmektedir.

Şekil 3.4. TCS34725 renk sensörü

Raspberry Pi üzerinde renk sensörü kullanılarak kırmızı, yeşil ve maviye karşılık gelen 0 ile 255 değerleri arasında 3'er değer gürültü kaynağı olan Tesla küresi içinde oluşan rastlantısal elektriksel ışımalardan elde edilecektir. Örnek olarak elektronik ortamda beyaz rengi (255,255,255) iken siyah rengin değeri ise (0,0,0)'dır. Python programlama dilinde TCS34725 sensörü yardımıyla renk değerleri;

```
import time
import board
import adafruit_tcs34725

i2c = board.I2C()
sensor = adafruit_tcs34725.TCS34725(i2c)

while True:

    color = sensor.color
    color_rgb = sensor.color_rgb_bytes
    print("#{0:02X} RGB Değeri: : {1}".format(color, color_rgb))

    temp = sensor.color_temperature
    lux = sensor.lux
    print("Sıcaklık: {0}K Lux Değeri: {1}\n".format(temp, lux))
    time.sleep(0.5)
```

komutları kullanılarak elde edilir.

3.1.4. Webcam Sensörü

Analog ve sayısal olmak üzere iki farklı tipi olan kameralar, temel video ve görüntü cihazı olarak dünyadaki birçok alanda yaygın olarak kullanılmaktadır. Uygulamalarından biri ise nesnelerin konumunu bularak çeşitli sonuçların elde edilmesidir. Bir nesnenin sahne içerisindeki konumu kamera tarafında birçok görüntü işleme yöntemleriyle elde edilebilmektedir, bu projede OpenCV kütüphanesi kullanılarak Şekil 3.5'de bulunan 1080p webcam ile hareket eden nesnenin konum değerlerinin elde edilmesi sağlanacaktır.

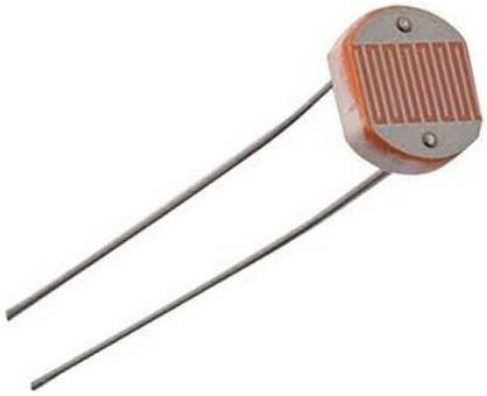
Şekil 3.5. 1080p webcam sensörü

Video işleme, bir görüntü karesi içerisindeki arka plan veya ön planda bulunan durağan ya da hareketli nesneler arasındaki etkileşimle ilgilenen bir kavramdır. Raspberry Pi cihazına USB portu ile takılan kamerayı kullanarak Tesla küresindeki elektriksel ışımaların hareketleri algılanıp x ve y koordinat değerleri ile ham veriler elde edilecektir. Kameradan alınacak görüntünün yenileme hızının yüksek bir seviyede olması maksadıyla 600x600 çözünürlük değerinde veriler elde edilip konum değerlerinin çıktısı alınmıştır. Görüntüdeki yenileme hızının yüksek olmasıyla birim zamanda oluşacak hareket daha fazla olacağından, belli bir zaman aralığında rastgele sayı üretimi maksadıyla fazla sayıda tohum değeri çıktısı oluşacaktır. Elde edilecek rastgele sayıların oluşma sürelerinin kısa olması, rastgelelik testleri için ihtiyaç olan uzun bit dizilerinin daha erken zamanda oluşmasını sağlayacaktır.

3.1.5. Işık Sensörü

LDR (Light Dependent Resistor–Işığa Bağımlı Direnç) birçok kişi tarafından fotosel olarak bilinen, Şekil 3.6'da gösterilen ışığın şiddetini ölçmeye yarayan (Salim vd., 2015) iki uçlu bir sensördür. Üzerine düşen ışığın miktarına göre direncin değeri değişerek ışığın değeri ölçülmektedir. Ardunio üzerinde bu sensör kullanıldığı zaman

analog girişe bağlanarak değerler elde edilir. Ancak Raspberry Pi'de analog giriş mevcut olmadığından bir RC devresinin kurulması gerekmektedir. RC devresi direnç ve yükü depolamaya yarayan kapasitörden oluşmaktadır. Akım kapasitörü şarj eder, direnç ise kapasitöre giden akımı düzenleyerek şarj olma süresini yavaşlatır. Bu sayede direncin değeri, kapasitörün dolacağı süreyi belirlemektedir. LDR sensörünün direnci üzerine düşen ışık miktarına göre değişeceğinden, kapasitörün şarj dolma süresi ışığın lux cinsinden değerini belirleyecektir.



Şekil 3.6. 10 mm LDR ışık sensörü

Raspberry Pi üzerinde ışık sensörü kullanılarak lux cinsinden değerler, gürültü kaynağı olan Tesla küresi içinde oluşan rastlantısal elektriksel ışımalardan elde edilecektir. Elde edilen değerlere (1094, 638, 1102, 1061,…) örnek olarak verilebilir. Python programlama dilinde LDR ışık sensörü yardımıyla ışık değerleri;

```
import time
import RPi.GPIO as GPIO
import hashlib
import timeit

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
sayac = 0

def RCtime (RCpin):

   reading = 0
   GPIO.setup(RCpin, GPIO.OUT)
   GPIO.output(RCpin, GPIO.LOW)
   time.sleep(.1)

   GPIO.setup(RCpin, GPIO.IN)
```

```
    while (GPIO.input(RCpin) == GPIO.LOW):
        reading += 1
    return reading

while(sayac < 5000):

    LDRReading = RCtime(3)
    print (LDRReading)

    print (LDRReading%16)
    ldr_mod16 = LDRReading%16
    if ldr_mod16 == 0:
        key = str(0)
    else:
        key= hex(ldr_mod16).lstrip('0x')
    print ('On altılık tabandaki karşılığı: ', key)
    sayac = sayac + 1
    print('Sayaç = ', sayac)
    f = open('/home/pi/FIPS/RNGMOD16/RNGLDRMOD16_20k.txt','a')
    f.write(key)
    time.sleep(0.1)
```

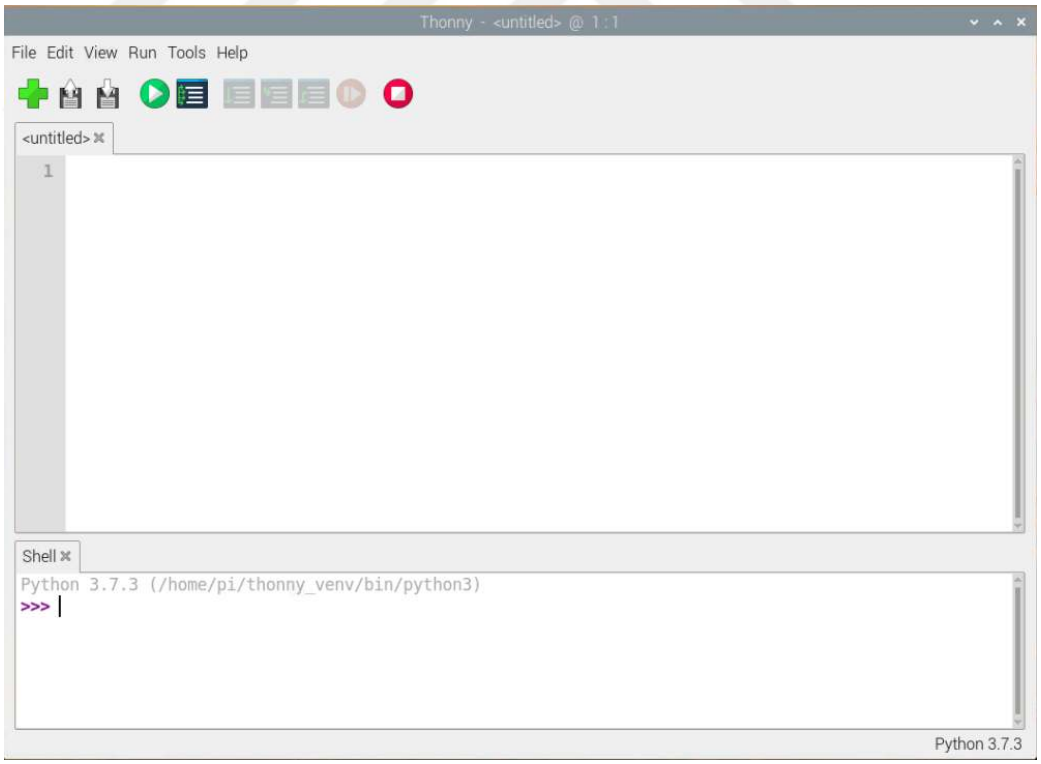
komutları kullanılarak elde edilir.

3.1.6. Raspbian İşletim Sistemi

İşletim sistemleri, donanım ve yazılım kaynaklarını yöneterek bilgisayar uygulamaları için ortak hizmetler sağlayan bir yazılım bileşenidir. Raspbian, Raspberry Pi için tasarlanmış Debian tabanlı bir işletim sistemidir (Le vd., 2022). Raspbian işletim sistemi üzerinde Python, C/C++ ve Node.js programlama dilleri ile geliştirme yapılabilir (Kour ve Arora, 2020). 2012 yılında piyasaya sürülen bu işletim sistemi, günümüze kadar gelişmeye ve kullanımı artmaya devam etmiştir (Sainz-Raso vd., 2019). İşletim sisteminin kurulumu için; 4 GB ve üzeri hafıza kartı, Raspberry Pi, Raspbian imaj dosyası ve imaj dosya yazıcı gereklidir. İlk olarak Raspbian işletim sistemi masaüstü bilgisayara indirilir. Daha sonra SD kart formatlandıktan sonra indirilen işletim sisteminin imaj dosyası kurulumu bu karta yapılır. Bu aşamadan sonra işletim sistemi kurulumu tamamlanmış olan Raspberry Pi mini bilgisayarı, projeler geliştirmeye hazır hale gelmiş olur.

3.1.7. Python Programlama Dili

Son zamanlarda kullanımı artan, genel amaçlı üst seviyeli bir programlama dilidir (Srinath, 2017). Temel felsefesi, geleneksel programlama dillerine göre mümkün olandan daha az sayıda kod satırı ile programın kısa şekilde kodlanmasıdır. Python ile kullanıcılar hem küçük hem de büyük çaplı projeler geliştirebilirler. Geniş bir kullanım alanına sahip, dinamik, nesneye yönelik ve genel amaçlı bir programlama dilidir. Hatalar diğer programlama dillerine göre daha fazla görmezden gelindiği için sonuç odaklı, esnek bir yapıya sahiptir. Çok basit ve anlaşılır bir sözdizimi olduğundan C++, Java ve C# programlama dillerine göre öğrenimi daha kolaydır. Windows, Linux ve Mac OS X gibi farklı işletim sistemlerinde (Sharma vd., 2020) bu programlama diliyle yazılan programlar, herhangi bir değişiklik yapmadan diğer platformlara taşınabilmektedir. Açık kaynak kodlu olan Python programlama dilinin değişiklik ve dağıtımında herhangi bir sınırlama yoktur. Raspberry Pi bilgisayarına Rasbian işletim sistemini kurulduktan sonra içerisinde gömülü olan Thonny Python Geliştirme Ortamı (derleyicisi) üzerinde proje geliştirilecektir. Thonny Geliştirme Ortamının arayüzü Şekil 3.7'de gösterilmiştir.



Şekil 3.7. Thonny geliştirme ortamı

3.1.8. OpenCV Kütüphanesi

Open Source Computer Vision (OpenCV) 1999 yılında Intel tarafından ortaya çıkarılan (Akkoca ve Buluş, 2021) açık kaynak kodlu olan ve ücretsiz olarak kullanılabilen (Nagpal,vd., 2018) bir görüntü işleme kütüphanesidir. İlk olarak C programlama dilinde, daha sonradan ise C++, Python, Java, C# gibi birçok dilde uygulamalar geliştirilmeye başlanmıştır. Platformdan bağımsız bir kütüphane olduğundan Linux, Windows, Android ve iOS gibi farklı işletim sistemi altyapılarında çalışabilmektedir. Bu kütüphane içerisinde yüz tanıma, hareket algılama, plaka tanıma, görüntü analizi, nesne ayırt etme gibi birçok algoritma vardır. OpenCV kütüphanesinin kaynak sayısının fazla olması ve geliştiriciler tarafından kolay erişilmesinden dolayı en fazla tercih edilen görüntü işleme teknolojisidir. Raspberry Pi ile Python programlama dili kullanılarak bu kütüphaneden faydalanılacaktır. Aşağıdaki iki komut çalıştırılarak OpenCV ve Numpy kütüphaneleri yüklenmiş olur:

Pip install opencv-python

Pip install numpy

Bu projede kullanılan OpenCV kütüphanesi ile; hareket algılanan alanın etrafına yeşil renk ile kutu çizilecek, çizilen kutunun ağırlık merkezinin koordinatları belirlenecek ve elde edilen bu x ve y değerleri rastgele sayı üretiminin tohum değeri olarak kullanılacaktır.

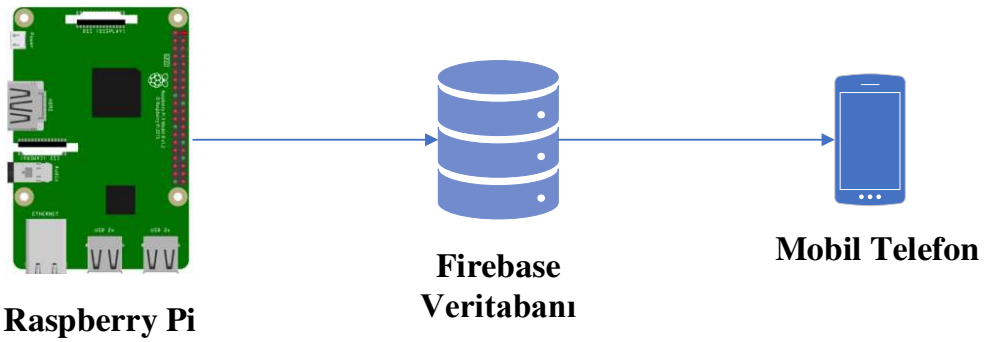
3.1.9. Firebase Gerçek Zamanlı Veritabanı

Firebase, verileri bulutta barındıran gerçek zamanlı bir veritabanıdır (Li vd., 2018). Veriler JSON (JavaScript Object Notation) olarak depolanır ve bağlı olduğu bütün istemcilerle gerçek zamanlı olarak (Khawas ve Shah, 2018) senkronize edilir. JSON veritabanı, farklı veri türlerini depolamada esneklik sunan ve veri modeli veya projede oluşabilecek değişiklikleri kolayca karşılayan NoSQL veritabanının bir parçasıdır. Firebase veritabanı, geleneksel http istekleri yerine veri senkronizasyonunu kullanır, böylece veride oluşabilecek her türlü değişiklikte bağlı olduğu cihaz bu güncellemeyi anlık olarak alır. Bu veritabanında veriler diskte saklandığı için Firebase uygulamaları çevrimdışı iken bile yanıt vermeye devam eder. Bağlantı tekrardan kurulduğu anda istemci cihaz kaçırdığı değişiklikleri alarak mevcut durumuyla senkronize edilir. Doğrudan mobil cihaz veya web tarayıcısı üzerinden Firebase veritabanına erişilebilir. Veriler okunurken veya yazılırken Firebase Gerçek Zamanlı Veritabanı Güvenlik

Kurallarına göre işlenmektedir. Şekil 3.8'de Firebase Gerçek Zamanlı Veritabanı arayüzü ile Tesla küresinden webcam sensörü ile anlık alınan x ve y koordinat verileri gösterilmiştir.

Şekil 3.8. Firebase gerçek zamanlı veritabanı arayüzü

Raspberry Pi ile anlık alınan x ve y koordinat verileri, Python programlama dili ile Firebase Gerçek Zamanlı Veritabanına anlık olarak depolanacaktır. Tasarlanan android tabanlı mobil uygulama ile veritabanına depolanmış olan veriler Şekil 3.9'da gösterilen genel yapıdaki sırada olacak şekilde mobil telefonda anlık olarak izlenecektir.

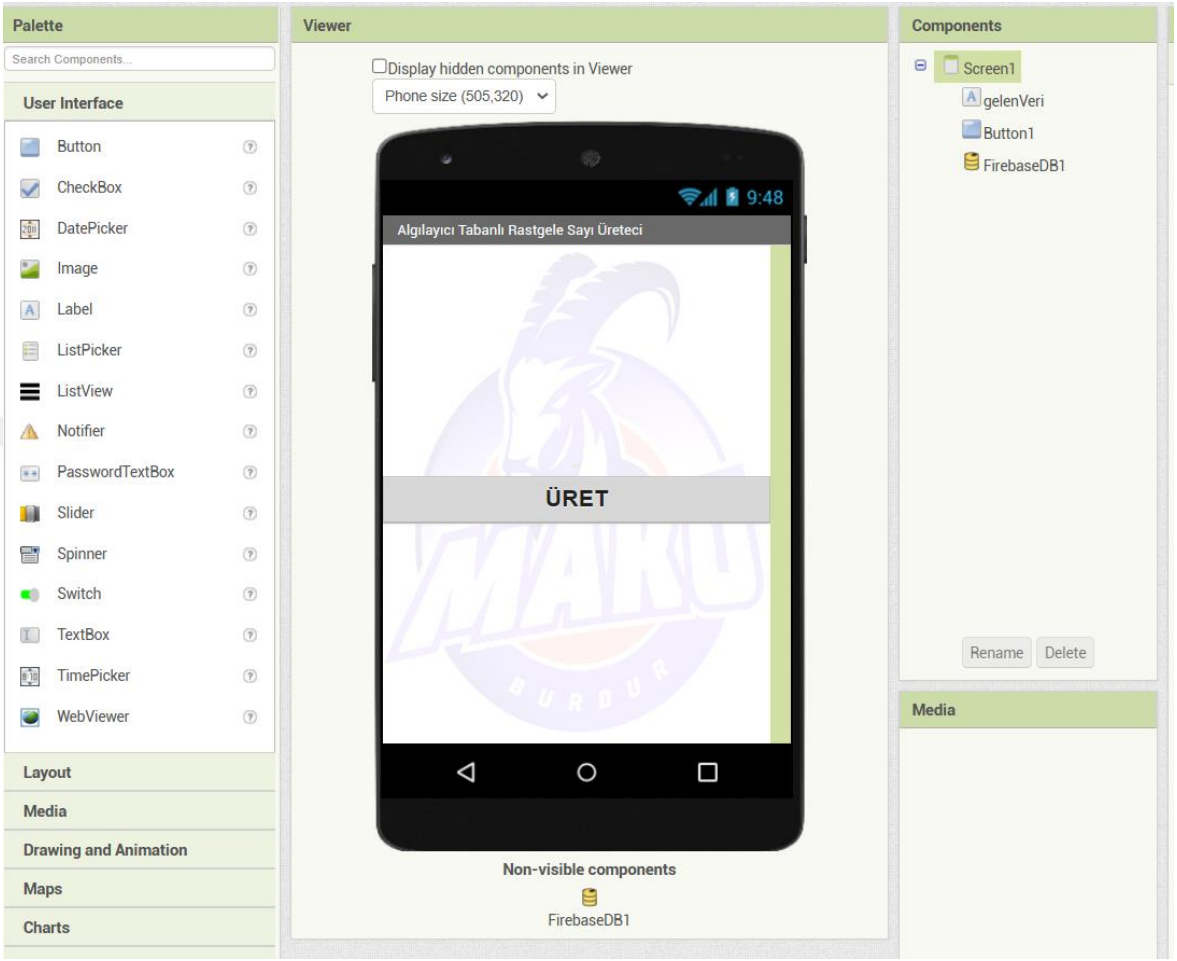


Şekil 3.9. Genel sistem yapısı

3.1.10. MIT App Inventor Mobil Uygulama Geliştirme Ortamı

MIT App Inventor Android sisteminde mobil uygulamalar geliştirme amacıyla oluşturulmuş, kullanıcılara basit bir arayüz sağlayan, dünya çapında milyonlarca kullanıcısı bulunan ve bir dizi bileşenlere işlevsellik sağlayan bir platformdur (Xie ve Abelson, 2016). Uygulamaların, görsel nesneleri sürükle-bırak mantığıyla oluşturulmasına olanak sağlayan (Mir ve Llueca, 2020) açık kaynaklı kodlu bir

arayüzdür. İnsanlar Raspberry Pi ve Arduino mini bilgisayarlarının sensörleriyle elde ettikleri verileri, MIT App Inventor ile geliştirdikleri uygulamalar üzerinden anlık olarak izleyebilmektedir. Tasarlanan sistemin arayüzü Şekil 3.10'da gösterilmiştir.



Şekil 3.10. MIT app inventor arayüzü

Tasarlanan bu sistem ile Raspberry Pi ile elde edilen ham veriler, Firebase Gerçek Zamanlı Veritabanına yüklenmekte ve MIT App Inventor ile tasarlanan Android tabanlı mobil uygulama üzerinden lokasyon bağımsız olarak anlık izlenebilmektedir. Veritabanı gerçek zamanlı olduğundan, sensörden elde edilen her yeni veri anlık olarak eski değerin üzerine kaydedilmekte ve yeni değer mobil uygulamada gösterilmektedir.

3.2. Rastgele Sayı Üreteçlerinin Genel Sınıflandırılması

Rastgele sayı üreteçleri aşağıda belirtilmek üzere sözde ve gerçek olarak iki farklı şekilde sınıflandırılmaktadır.

3.2.1. Sözde Rastgele Sayı Üreteçleri

Sözde Rastgele Sayı Üreteçleri, rastgele sayıyı deterministik olarak üreten sistemlerdir. Gerçek RSÜ'lerle karşılaştırıldığında, kolay oluşturulma ve düşük maliyetle üretilme gibi avantajları bulunmaktadır. Fakat kullanılan algoritmaların deterministik olmalarından dolayı, çıkışta üretilen sayıların rastgeleliği beklenildiği şekilde değildir. Algoritma ele geçirildiğinde, herhangi bir andaki değerine bakarak sonraki çıkışlar tahmin edilebilmektedir (Demirkol, 2007). Bu tahmin edilme durumu, gizlilik isteyen güvenli haberleşme sistemlerinde ciddi güvenlik sorunları oluşturabilmektedir (Huang vd., 2020). Kısaca gerçek RSÜ'lerle karşılaştırıldığı zaman istatistiksel olarak başarımı daha düşük RSÜ'lerdir.

Sözde RSÜ'ler tohum (seed) adı verilen bir başlangıç değerine bağlı olarak üretilirler. Bu başlangıç değeri; kullanıcı kaynaklı fare ya da klavye hareketleri, mikrofon ya da hoparlörün ses değerleri, işlemciden veya bellekten alınan ham verilerden elde edilebilir. Belirlenen başlangıç tohum değeri, son işlem algoritmalarına tabi tutularak çıkışta sabit uzunlukta rastgele bit dizileri üretir. Sözde RSÜ'leri hızlı üretilmeleri ve ucuz olmalarına karşın, periyodikleştikten sonra ya da tohum değerinin korunamamasından dolayı üretilen rastgele sayıların tamamen tahmin edilebilir olmasını mümkün kılabilir. Bu sebepten dolayı sözde RSÜ'lerin kritik olarak görülen kriptografik sistemlerde kullanımı önerilmemektedir (Stipčević ve Koç, 2014). Sözde RSÜ'lerde olması gereken bazı özellikler;

İlişkisiz Diziler; diziler arasında bir doğrudan ilişki olmadığını gösterir. Belirli bir sayı aralığı kullanılarak gelecek bitlerin tahmin edilmesi imkânsız olmalıdır. Kelimenin tam anlamıyla, çıkış bitlerinin herhangi iki bölümü arasında doğrudan ilişki olmamalıdır.

Uzun Periyot; ideal olarak, bitlerin art arda gelmesinde asla tekrar eden bir model olmamalıdır. Ancak, eninde sonunda bir tekrar oluşsa bile bu oluşumun mümkün olduğundan daha uzun bir zaman diliminde olması beklenmektedir. Başka bir deyişle, tekrar eden dizilerin oluşacağı kabul edilmektedir, ancak bunlar mümkün olduğunca aralıklı olmalıdır.

Tekdüzelik; sözde rastgele sayıların ikili gösterimleri en sık kullanılanlardır. Belirgin bir şekilde dağıtılmamaları gerekse de eşit miktarda 1 ve 0 olmalıdır. Rastgele sayılar serisi tutarlı ve tarafsız olmalıdır. 0 sayısına göre önemli ölçüde daha fazla veya önemli ölçüde daha az 1 varsa çıktının rastgele olduğundan bahsedilemez.

Hesaplamalı Ayırt Edilemezlik; belirli bir Sözde RNG'nin çıktısından elde edilen sayıların hiçbir alt bölümü, herhangi bir etkili yöntemle polinom zamanda diğer sayı alt bölümlerinden ayırt edilememelidir.

3.2.2. Gerçek Rastgele Sayı Üreteçleri

Gerçek RSÜ'ler, kaotik tabanlı gerçekleşen doğa olaylarının rastgeleliğini kullanıp, bir algoritmayla son işlem uygulayarak sayı üreten sistemlerdir. Çıkışta gerçek rastgele sayılar üretildiğinden, kritik şifreleme sistemlerinde çokça kullanılabilmektedir. Gerçek RSÜ'ler, deterministik olmayan fiziksel olayları entropi kaynağı olarak kullanarak, elektriksel ışımalar, ışık şiddeti, sıcaklık bilgisi gibi fiziksel süreçleri kaynak olarak alan gerçek rastgele diziler oluşturmaktadır. Entropi kaynağından elde edilen analog sinyaller sayısallaştırılarak örnekleme yapılır. Elde edilen bu sinyaller, günümüzde kullanılan IoT cihazları ve sensörlerinden elde edilebilir. Örneğin bir hayvanın ayağına takılan ölçü sensöründen alınan verilerle ya da tarım alanında bir bitkinin uzaktan izlenmesiyle çekilen istatistiki verilerle tahmin edilemeyecek rassal ham veriler elde edilebilir. Örnekleme sürecinden sonra zayıf istatistiki özellikler gösteren sayılar son işleme tabi tutularak, daha güçlü istatistikler göstermesi sağlanır (Yosunlu ve Avaroğlu, 2020).

Entropi kaynağından elde edilip son işlem yapılarak elde edilen veri kümeleri, gerçekliğe yakın olmalarının kontrol edilmesi maksadıyla rastgelelik testlerine tabi tutulur. Eğer üretilen sayılar gerçekten rastgele ise, bu testlerden başarılı bir şekilde geçer. Üretilen sayılar bir veya birden fazla teste sokulup sonuçlar analiz edilebilir. Bu tezde Raspberry pi cihazına bağlı bazı sensörler yardımıyla fiziki dünyadan elde edilen sayılar testlere sokulmuş, elde edilen test sonuçları karşılaştırılmıştır. Literatürde yapılan rastgelelik testlerden bazılarına; NIST, FIPS 140, Diehard, Ki kare, Kolmogorov-smirnov olarak örnek verilebilir. Bu testlerde genellikle; dizideki bir ve sıfır rakamlarının oranı, bir ve sıfır rakamlarının M bitlik bloklara bölünerek oranının hesaplanması, art arda gelen bir ve sıfırların sayısı, tekrar eden blokların tespiti gibi analizler yapılmaktadır. Bu projede, farklı algılayıcı kaynaklar (kamera ve ışık sensörü) ve yöntemlerle (mod yöntemi, son bitlerin elde edilmesi ve özetleme algoritmaları) elde edilen bit dizilerinin monobit ve poker test sonuçları karşılaştırılarak değerlendirilmiştir.

3.3. FIPS 140-1 Test Paketi

FIPS 140-1 test paketi; monobit, poker, run ve long run olmak üzere 4 adet adımdan oluşmaktadır. Elde edilen dizilerin rastgele olarak adlandırılabilmesi için söz konusu

testlerin tümünden geçmesi gerekmektedir. FIPS 140-1 testinin uygulanabilmesi için 20.000 uzunluğundaki bit dizisi elde edilmesi gerekmektedir.

3.3.1. Monobit Testi

Monobit testinde, bir dizideki birlerin sayısının sıfırlara göre oranı kıyaslanır. Dizi uzunluğunun (n) olduğu bir veri kümesinde, birlerin sayısının n/2 adete yakın olması, üretilen dizinin kalitesi hakkında bilgi verir. 20 bin bit uzunluğundaki bir dizi için birlerin sayısının 9.654-10.346 arasında olması testin başarıyla sonuçlandığı, birlerin sayısının bu aralıkta olmaması ise testin başarısız olduğu anlamına gelir (Luengo vd., 2022). Bu makalede gürültü kaynağı olarak Tesla küresinden kamera ve ldr sensörü ile ham veri dizileri elde edilecektir. Ham veriler kullanılarak 4 farklı yöntem ile oluşturulan 1.024 bit uzunluğundaki çıkışlar monobit testine, 20.000 bit uzunluğundaki çıkışlar ise poker testine sokulup sonuçları analiz edilecektir.

3.3.2. Poker Testi

Poker testinde üretilen 20.000 bitlik rastgele dizi, dörder bitlik bloklara ayrılarak toplamda 5.000 adet sayı elde edilir. Bu dört bit ile onaltılık tabanda sayılar ifade edilmektedir. Üretilen sayıların rastgele olabilmesi ve güvenilirliğinin kanıtlanabilmesi için 16 farklı sayının her birinin eşit sayıda olması beklenir. Aşağıdaki formül ile hesaplanan poker değerinde; i sayısı sıfırdan on beşe kadar olan dört bitlik sayıları, f(i) ise i sayısından kaç tane olduğunu gösterir. Testten başarılı bir şekilde geçilebilmesi için hesaplanan poker değerinin 1,03 ile 57,4 arasında olması gerekmektedir.

i: 0'dan 15'e kadar olan dört bitlik sayılar

$f(i)$: i sayısının adedi

$$\text{Poker değeri} = \frac{16}{5000} * \left(\sum_{i=0}^{15} [f(i)]^2 \right) - 5000 \quad (3.1)$$

3.3.3. Run Testi

Elde edilen rasgele sayıların bu testten geçebilmesi için dizi içerisindeki art arda gelen 0 ve 1 değerlerinden oluşan farklı uzunluktaki bit değerlerinin Çizelge 3.1'de belirlenen aralıklar içerisinde olması gerekmektedir. Örneğin Çizelge 3.1'de gösterildiği gibi blok uzunluğunun 3 olması durumunda $502 \leq x \leq 748$ şartının sağlanması gerekir. Burada "x" ile ifade edilen bit dizisinin uzunluğudur.

Çizelge 3.1. Run testi blok uzunlukları (Bakır, 2017)

Blok Uzunluğu	Blok Sayısı Aralığı
1	$2267 \leq x \leq 2733$
2	$1079 \leq x \leq 1421$
3	$502 \leq x \leq 748$
4	$223 \leq x \leq 402$
5	$90 \leq x \leq 223$
6+	$90 \leq x \leq 223$

3.3.4. Long Run Testi

Run test ile benzer olan bu testin tek farkı, 20.000 bit serisindeki 0 ve 1'lerden oluşan blokların toplam sayısının 34'ten küçük olmasıdır. Eğer bu şart gerçekleşmezse test başarısız sayılacaktır.

3.4. NIST 800-22 Test Paketi

NIST tarafından oluşturulan NIST 800-22 testinin on beş farklı aşaması vardır ve yaygın olarak kabul görmektedir. Kontrol edilen sayının bit uzunluğu en az 1.000.000 bit olmalıdır. Testin başarılı sayılabilmesi için testin her aşamasında üretilen P değerinin $P \geq 0{,}01$ gereksinimini karşılaması gerekir. P değerinin 0 olmasıyla anahtar dizisinin rasgele olmasından söz edilemez iken 1'e yaklaşmasıyla anahtar dizisinin mükemmel rastgelelik seviyesinde olduğundan bahsedilir. Testin bir aşaması bile başarısızlıkla sonuçlanırsa, rastgele sayı üretiminin başarısız olduğu söylenir. NIST 800-22 testinin alt adımları ve P değerinin nasıl hesaplanacağına ilişkin bilgiler aşağıdaki bölümlerde açıklanmaktadır.

3.4.1. Frekans Testi

Bu test, rastgele veri akışında 0 ve 1 sayılarının ne sıklıkta göründüğüne bakar. Bu oran yaklaşık olarak aynı olmalıdır. Sonraki tüm testler için bu testi geçmek bir ön koşuldur.

3.4.2. Blok Frekans Testi

Blok Frekans testi, tüm bit dizisine bakan Frekans testinin aksine, m bit blok uzunluğuna sahip rastgele bir dizideki 0'ların ve 1'lerin sayısını incelemektedir. Test edilecek dizinin bit uzunluğu en az $m = 100$ olmalı ve blok uzunluğunun $n = 20$ olduğu varsayılmalıdır. M bitlik bir blokta m/2 oranında 1 bulunması, bu testi geçmek için ön koşuldur.

3.4.3. Akış Testi

Test edilen bit dizisinde, bu test ardışık 0 ve 1 bloklarını arar ve 1 değerleri kademeli ve hızlı değişiklikler açısından değerlendirilir.

3.4.4. Blok İçinde En Uzun Bit Yenilemesi Testi

Bu testte, incelenecek bloklar içerisindeki ardışık olan en uzun 1 değerlerine bakılır. Dizi uzunluğunun n bit olduğu durumda, m blok uzunluğu Çizelge 3.2'de gösterildiği gibi olmalıdır.

Çizelge 3.2. Diziye göre blok uzunluğu (Bakır, 2017)

Minimum n	m
128	8
6.272	128
750.000	10.000

3.4.5. İkili Matris Rank Testi

Bu testte orijinal rastgele sayı dizisi ve bu dizinin sabit uzunluktaki alt dizileri kullanılır. Alt diziler arasında doğrusal bağımlılık olup olmadığı kontrol edilir.

3.4.6. Ayrık Fourier Dönüşümü Testi

Bu test, dönüşümdeki ayrık Fourier tepe noktalarını inceler. %95'lik tepe noktasını geçen tepe sayısının %5'e göre hangi oranda farklı olduğunu belirlemektir.

3.4.7. Örtüşmeyen Şablon Testi

Bu test, n bit veri uzunluğuna sahip bir rastgele sayı dizisinde, seçilen m bitlik bloklar içinde belirli periyodik örnek dizilerinin oluşma sıklığını ölçmektedir. Örnek dizisi bloklarının tekrarlanması durumunda bir sonraki bloğun ilk bitinden devam ettirilir. Örnek dizi blokları bulunmazsa pencere bir bit ötelenerek arama tekrar gerçekleştirilir.

3.4.8. Örtüşen Şablon Eşleştirme Testi

Bu test örtüşmeyen şablon testi ile benzer şekilde gerçekleştirilmektedir. Farklı olarak örnek dizi bloklarının tespit edilmesi durumunda pencere bir bit ötelenerek arama tekrar gerçekleştirilir.

3.4.9. Maurer'in Evrensel İstatistik Testi

Bu testte birbiriyle eşleşen modellerin bit uzunluğuna bakılmaktadır. Buradaki amaç herhangi bir kayba neden olmadan dizinin sıkıştırılıp sıkıştırılmayacağının analiz edilmesidir. Eğer dizi önemli bir ölçüde sıkıştırılıyor ise bu dizinin rastgele olmadığı sonucuna varılmaktadır.

3.4.10. Doğrusal Karmaşıklık Testi

Bu testte doğrusal geri besleme kaydırma yazmacının uzunluğunun kaç olduğuna bakılmaktadır. Doğrusal geri besleme kaydırma yazmacının çok kısa olması rastgele olmama, uzun olması ise rastgeleliği sağlama anlamına gelmektedir.

3.4.11. Seri Testi

Bu testte üretilen dizi içerisindeki m bitlik blokların diğer m bitlik bloklar ile bir benzerliğin olup olmadığı incelenmektedir. Bu test seri-1 ve seri-2 olmak üzere iki ayrı aşamada gerçekleşmektedir.

3.4.12. Yaklaşık Entropi Testi

Seri testinde olduğu gibi bu testte de m bit uzunluğundaki blokların frekanslarına bakılır. Ardışık olan iki bloğun frekansı ile rastgele sayı dizisi için beklenen frekans değeri karşılaştırılmaktadır.

3.4.13. Kümülatif Toplamlar Testi

Rastgele olarak üretilen dizi bloklara ayrılıp 0 ve 1 sayılarının denge oranı belirlenerek bloklar arasındaki denge farkına bakılmaktadır.

3.4.14. Rastgele Gezinimler Testi

Bu testte dizi içindeki ardışık olan bloklarda bulunan 0 ve 1 sayılarının oranı belirlenmektedir. Oran belirlendikten, sonra bloklar arasında bu oranın dağılımı incelenir.

3.4.15. Rastgele Gezinimler Değişken Testi

Rastgele gezinimler testinde olduğu gibi bu testte de dizi içindeki ardışık olan bloklarda bulunan 0 ve 1 sayılarının oranı belirlenmektedir. Ortalama değere göre bu oranın sapma miktarına bakılmaktadır.

3.5. Yöntem

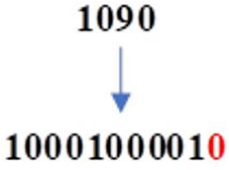
Bu bölümde kullanılan 4 ayrı yöntemin örneklerle açıklaması verilmiştir.

3.5.1. Son Bitler Yöntemi

Son bitler yöntemi, koordinat ya da ışık değerlerinin ikili sayı sistemine çevrilerek sondaki bitinin alınmasıyla gerçekleşir. Webcam sensörüyle her çevrimde x ve y olarak iki farklı onluk sayı sistemdeki değer elde edilecek ve elde edilen değerler ikili sayı sistemine çevrilerek sondaki birer adet bit alınarak rastgele sayı üretimi için dizi oluşturulacaktır. Örneğin webcam sensörüyle (197 ve 121) değerlerinin son bitler yöntemiyle ikili sayı sitemine çevrildikten sonra (11000101 ve 01111001) elde edilen son bit değerleri iki koordinat verisi için de Şekil 3.11'de görüldüğü gibi '1' dir.

Şekil 3.11. Webcam ile son bitler yöntemi

Webcam sensörünün bir hareketi algılamasıyla oluşan x ve y koordinatları bu yöntem ile elde edilen son bit değerlerine göre iki bit uzunluğunda bir çıkış oluşturur. İstenilen sabit anahtar uzunluğu oluşuncaya kadar hareket algılama döngüsü devam eder. Webcam sensörüyle 256 bit sabit uzunluğunda bir çıkış elde edilmesi için 128 çevrimin olması gerekmektedir. Işık sensörüyle her çevrimde onluk sayı siteminde tek değer elde edilecek ve elde edilen bu değer ikili sayı sistemine dönüştürülerek sondaki bir adet bit alınarak rastgele sayı üretimi için dizi oluşturulacaktır. Örneğin ışık sensörüyle (1090) değerinin son bitler yöntemiyle ikili sayı sistemine çevrildikten sonra (10001000010) elde edilen son bit değeri Şekil 3.12'de görüldüğü gibi '0' dır.

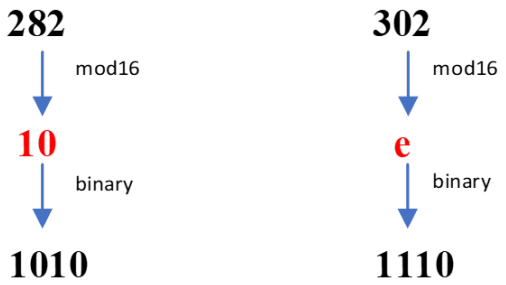


Şekil 3.12. Işık sensörü ile son bitler yöntemi

Işık sensörünün her bir çevrimi sonucu elde edilen son bit değerlerine göre bir bit uzunluğunda bir çıkış oluşturulur. Işık sensörüyle 256 bit sabit uzunluğunda bir çıkış elde edilmesi için 256 çevrimin olması gerekmektedir. İstenilen sabit anahtar uzunluğu oluşuncaya kadar ışık değeri algılama döngüsü devam eder.

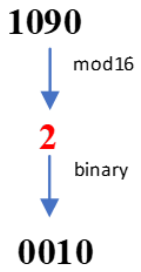
3.5.2. Mod 16 Yöntemi

Mod 16 yöntemi, koordinat ya da ışık değerlerinin 16 sayısına bölünmesiyle elde edilen kalanın (mod 16) onaltılık sayı sistemine çevrilmesiyle gerçekleşir. Webcam sensörüyle her çevrimde x ve y olarak iki farklı onluk sayı sisteminde değer elde edilecek ve 16 sayısına bölünmesiyle elde edilen kalanların onaltılık sisteme çevrilmesi sonucu rastgele sayı üretimi için dizi oluşturulacaktır. Örneğin webcam sensörüyle (282 ve 302) değerlerinin mod 16 yöntemiyle işlenmesi sonucu elde edilen kalanlar sırasıyla onaltılık sayı sisteminde Şekil 3.13'te görüldüğü gibi '10' ve 'e' dir.



Şekil 3.13. Webcam ile mod 16 yöntemi

Webcam sensörünün bir hareketi algılamasıyla oluşan x ve y koordinatları bu yöntem ile elde edilen kalan değerlerine göre sekiz bit uzunluğunda bir çıkış oluşturur (1010 1110). Webcam sensörüyle 256 bit sabit uzunluğunda bir çıkış elde edilmesi için 32 çevrimin olması gerekmektedir. İstenilen sabit anahtar uzunluğu oluşuncaya kadar hareket algılama döngüsü devam eder. Işık sensörüyle her çevrimde onluk sayı siteminde tek değer elde edilecek ve 16 sayısına bölünmesiyle elde edilen kalanın onaltılık sisteme çevrilmesi sonucu rastgele sayı üretimi için dizi oluşturulacaktır. Örneğin ışık sensörüyle (1090) değerinin mod 16 yöntemiyle işlenmesi sonucu elde edilen kalan onaltılık sayı sisteminde Şekil 3.14'de görüldüğü gibi '2' dir.



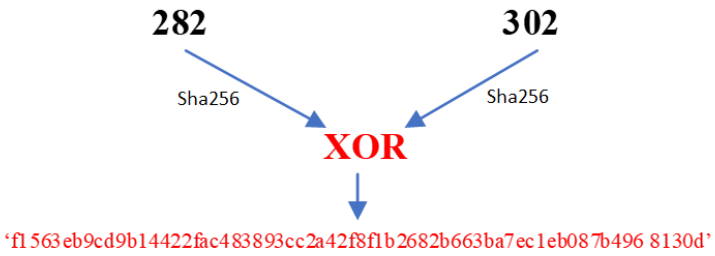
Şekil 3.14. Işık sensörü ile mod 16 yöntemi

Işık sensörünün her bir çevrimi sonucu elde edilen kalan değerine göre dört bit uzunluğunda bir çıkış oluşturulur (0010). Işık sensörüyle 256 bit sabit uzunluğunda bir çıkış elde edilmesi için 64 çevrimin olması gerekmektedir. İstenilen sabit anahtar uzunluğu oluşuncaya kadar ışık değeri algılama döngüsü devam eder.

3.5.3. Sha256 Yöntemi

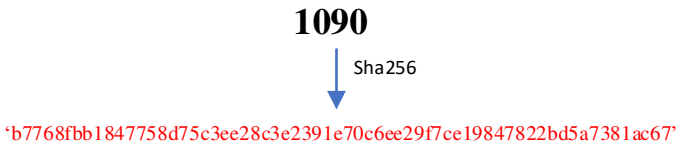
Ham bilgiyi, orijinal biçimine geri döndüremeyecek şekilde karıştırma işlemine özetleme (hashing) adı verilir. Girdi olarak alınan veri, matematiksel işlemlerden geçirilir ve elde edilen çıktı özet olarak isimlendirilir, bu özet ile verinin bütünlüğü sağlanmış olur. Çok büyük boyutlu verilerin özeti alınarak küçük boyutlu verilere indirgeme işlemi gerçekleşebilir. NIST tarafından geliştirilmiş Sha256 algoritması (Kasgar vd., 2012); giriş verisinin karakter uzunluğuna bakılmaksızın, çıkışta sabit 256 bit uzunluğunda, tanınmayan yeni bir veri dizisine dönüştüren özetleme algoritmasıdır. Giriş verisinde bir noktada bile değişiklik olursa çıkışta tamamen farklı özet elde edilir.

Sha256 yöntemi koordinat ve ışık değerlerinin özetleme algoritmasına sokulmasıyla gerçekleşir. Webcam sensörüyle her çevrimde x ve y olarak iki farklı onluk sayı sisteminde değer elde edilecek ve elde edilen bu değerler Sha256 özetleme algoritmasıyla işlenmesi sonucu elde edilen 256 bit uzunluğundaki iki değer XOR işlemine sokulması sonucu rastgele sayı üretimi için dizi oluşturulacaktır. Örneğin webcam sensörüyle (282 ve 302) değerlerinin Sha256 yöntemiyle işlenmesi sonucu elde edilen çıktıların XOR mantıksal işlemine sokulması sonucu Şekil 3.15'de görüldüğü gibi 'f1563eb9cd9b14422fac483893cc2a42f8f1b2682b663ba7ec1eb087b496 8130d' değeri elde edilmektedir.



Şekil 3.15. Webcam ile Sha256 yöntemi

Webcam sensörünün bir hareketi algılamasıyla oluşan x ve y koordinatları bu yöntem ile elde edilen değerlere göre 256 bit uzunluğunda bir çıkış oluşturur. Webcam sensörüyle 256 bit sabit uzunluğunda bir çıkış elde edilmesi için bir çevrimin olması yeterlidir. Bir çevrimde daha fazla bit çıktı verdiğinden dolayı mod 16 ve son bitler yöntemine göre daha hızlıdır. İstenilen sabit anahtar uzunluğu oluşuncaya kadar hareket algılama döngüsü devam eder. Işık sensörüyle her çevrimde onluk sayı sisteminde tek değer elde edilecek ve elde edilen bu değer Sha256 özetleme algoritmasıyla işlenmesi sonucu rastgele sayı üretimi için dizi oluşturacaktır. Örneğin ışık sensörüyle (1090) değerinin Sha256 yöntemiyle işlenmesi sonucu Şekil 3.16'da görüldüğü gibi 'b7768fbb1847758d75c3ee28c3e2391e70c6ee29f7ce19847822bd5a7381ac67' değeri elde edilmektedir.



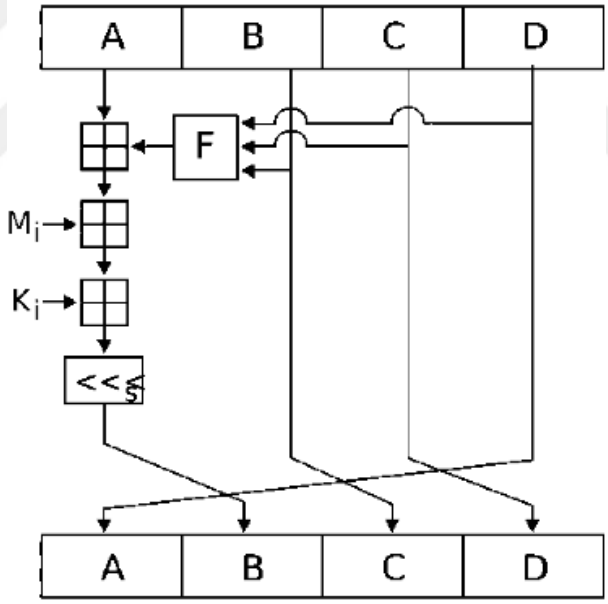
Şekil 3.16. Işık sensörü ile Sha256 yöntemi

Işık sensörünün her çevrimi sonucu elde edilen değere göre 256 bit uzunluğunda bir çıkış oluşur. Işık sensörüyle 256 bit sabit uzunluğunda bir çıkış elde edilmesi için bir çevrimin olması yeterlidir. Bir çevrimde daha fazla bit çıktı verdiğinden dolayı mod 16 ve son bitler yöntemine göre daha hızlıdır. İstenilen sabit anahtar uzunluğu oluşuncaya kadar ışık değeri algılama döngüsü devam eder.

3.5.4. Md5 Yöntemi

Md5 (Message Digest 5) özetleme algoritması ise Ron Rivest tarafından (Rachmawati vd., 2018) önceki bir hash fonksiyonu olan Md4 algoritmasının yerini alması için oluşturmuştur. Sha256'dan farklı olarak çıkışta 128 bit uzunluğunda sonuç verir. Tek yönlü olarak çalışan bu algoritmada, şifreleme yapılırken şifre çözümleme işlemi gerçekleştirilemez. Girdi olarak verilen mesaj on altı adet 32 bitlik kelimeler olacak şeklinde 512 bitlik segmentlere ayrılır. Md5'in amacı 128 bitlik bir çıktı üretmek olduğundan, algoritma Şekil 5.7'da görüldüğü gibi A, B, C ve D harflerini içeren dört adet 32 bitlik kelimeden oluşan 128 bitlik bir parça üzerinde çalışır. Şekil 3.17'de;

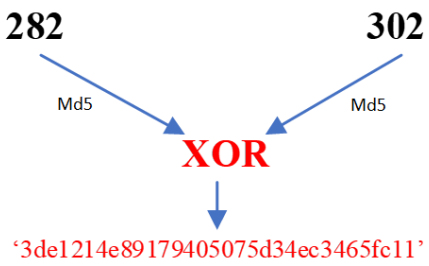
F: doğrusal olmayan bir fonksiyonu
Mi: mesaj girişinin 32 bitlik bloğunu,
Ki: her işlem için farklı olan 32 bitlik bir sabiti ifade etmektedir.



Şekil 3.17. Md5 genel yapısı (Ali ve Farhan, 2020)

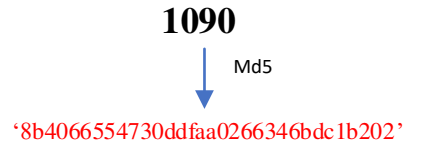
Md5 yöntemi, koordinat ve ışık değerlerini özetlemek maksadıyla kullanılmaktadır. Webcam sensörüyle her çevrimde x ve y olarak iki farklı onluk sayı sisteminde değer elde edilecek ve elde edilen bu değerler Md5 özetleme algoritmasıyla işlenmesi sonucu elde edilen 128 bit uzunluğundaki iki değer XOR işlemine sokulması sonucu rastgele sayı üretimi için dizi oluşturulacaktır. Örneğin webcam sensörüyle (282 ve 302) değerlerinin Md5 yöntemiyle işlenmesi sonucu elde edilen çıktıların XOR

mantıksal işlemine sokulması sonucu Şekil 3.18'de görüldüğü gibi '3de1214e89179405075d34ec3465fc11' değeri elde edilmektedir.



Şekil 3.18. Webcam ile Md5 yöntemi

Webcam sensörünün bir hareketi algılamasıyla oluşan x ve y koordinatları bu yöntem ile elde edilen değerlere göre 128 bit uzunluğunda bir çıkış oluşturur. Webcam sensörüyle 256 bit sabit uzunluğunda bir çıkış elde edilmesi için iki çevrimin olması yeterlidir. Bir çevrimde daha fazla bit çıktı verdiğinden dolayı mod 16 ve son bitler yöntemine göre daha hızlıdır. Sha256 yöntemine göre iki kat daha fazla çevrim yapması gerekmektedir. İstenilen sabit anahtar uzunluğu oluşuncaya kadar hareket algılama döngüsü devam eder. Işık sensörüyle her çevrimde onluk sayı sisteminde tek değer elde edilecek ve elde edilen bu değer Md5 özetleme algoritmasıyla işlenmesi sonucu rastgele sayı üretimi için dizi oluşturacaktır. Örneğin ışık sensörüyle (1090) değerinin Md5 yöntemiyle işlenmesi sonucu Şekil 3.19'da görüldüğü gibi '8b4066554730ddfaa0266346bdc1b202' değeri elde edilmektedir.



Şekil 3.19. LDR ile Md5 yöntemi

Işık sensörünün her çevrimi sonucu elde edilen değere göre 128 bit uzunluğunda bir çıkış oluşur. Işık sensörüyle 256 bit sabit uzunluğunda bir çıkış elde edilmesi için iki çevrimin olması yeterlidir. Bir çevrimde daha fazla bit çıktı verdiğinden dolayı mod 16 ve son bitler yöntemine göre daha hızlıdır. İstenilen sabit anahtar uzunluğu oluşuncaya kadar ışık değeri algılama döngüsü devam eder.

4. ARAŞTIRMA BULGULARI VE TARTIŞMA

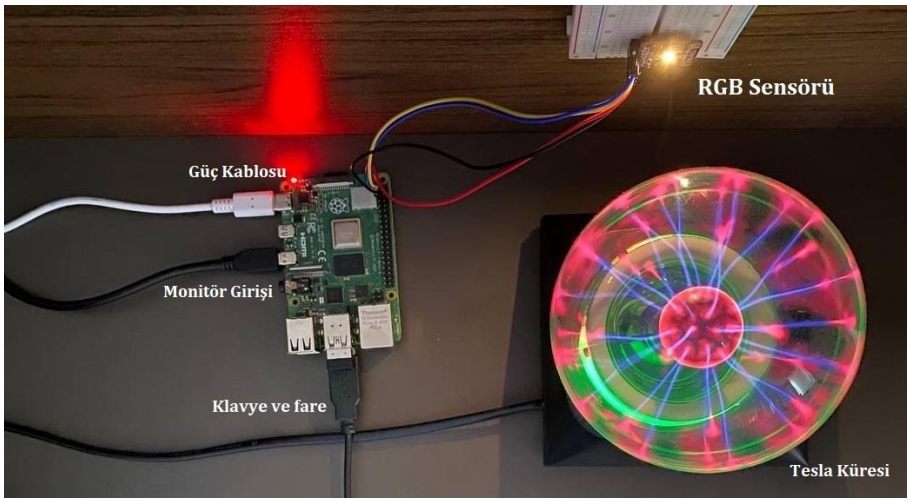
Bu bölümde projede kullanılan sensörlerden elde edilen veriler ve bu verilerin test sonuçları hakkında bazı bilgiler verilmiştir.

4.1. Sensörlerden Elde Edilen Veriler

Rastgele sayılar genellikle evrende bulunan doğa olaylarından elde edilen verilerle üretilmektedir. Bu doğa olaylarına; sıcaklık gürültüleri, zaman, elektriksel ışımalar, konum vb. kavramlar örnek verilebilir. Sıcaklık sensörü, ışık sensörü, kamera, renk sensörü, hız sensörü vb. IoT donanımlarıyla, doğa olaylarından alınan veriler sayısal değerlere dönüştürülür. Söz konusu olaylardan elde edilen değerlerle gerçek rastgele sayılar üretilebilir. Teorik olarak incelenirse doğa olayları birbirinden bağımsız olduğundan, üretilen sayılar arasında bir ilişki bulunmamaktadır.

4.1.1. RGB Sensöründen Elde Edilen Veriler

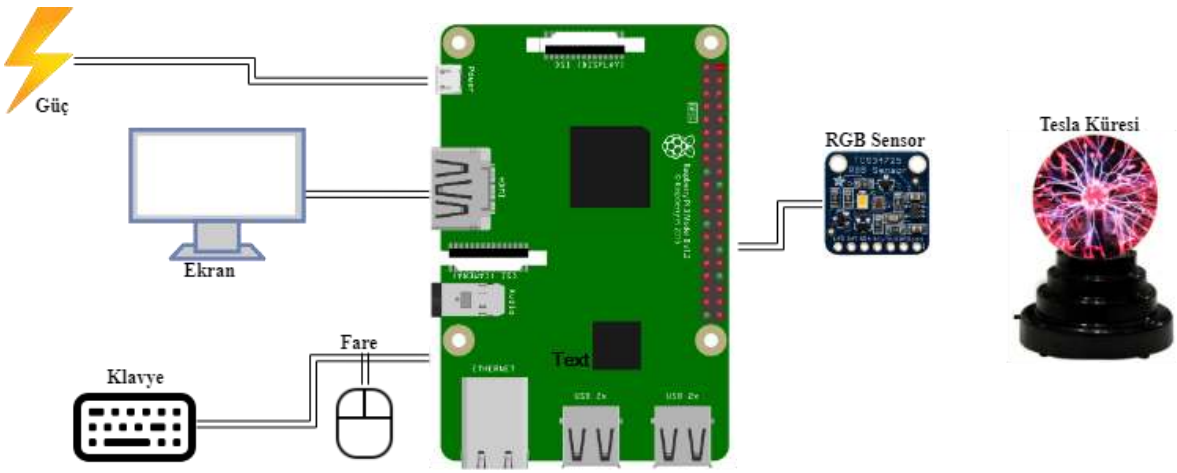
Bu bölümde Raspberry Pi cihazına bağlanan TCS34725 renk sensörünün entropi kaynağı olarak kullanılan Tesla küresinden elde ettiği ham veriler incelenecektir. Bu sensör renk değerleri haricinde renk sıcaklık ve renk ışık şiddeti değerlerini de elde etmektedir. Renk sensörleri kırmızı, yeşil ve mavi olan ana renklerin karışımıyla 0 ile 255 arasındaki renk değerlerine ulaşmayı hedefler. Bu sensörler; sensörden çıkan ışığın maddeye çarpması ve maddeden yansıyıp elde edilen ışık değerlerini karşılaştırarak sonuca ulaşma mantığıyla çalışır. Renk sensörü üzerinde bulunan GND, SCL, SDA ve 3V3 pinleri ile Raspberry Pi cihazında karşılık gelen pinler breadboard üzerinden erkek-dişi ara kablolar yardımıyla birbirine bağlanmıştır. RGB sensörü kullanılarak tasarlanan deney setinin genel görünüşü Şekil 4.1'de gösterilmiştir.



Şekil 4.1 RGB sensörlü sistemin genel görünüşü

Şematik gösterimi ise Şekil 4.2'de verilen sistem için kullanılan malzemeler aşağıda listelenmiştir:

- Raspberry Pi 4,
- TCS34725 RGB Renk Sensörü,
- Tesla Küresi,
- Monitör,
- Klavye ve fare.



Şekil 4.2. RGB sensörlü deney düzeneğinin şekli

Tesla küresinin merkezinde bulunan küçük bir küreden, dış kısmında bulunan cam küreye doğru farklı renklerde rastlantısal elektriksel ışımalar meydana gelir. Python dili ile renk sensöründen kırmızı, yeşil ve mavi olan 0 ile 255 değerleri arasında 3 değerin elde edilmesi maksadıyla kodlama işlemi yapılmıştır. Gürültü kaynağı olarak fiziksel

dünyadaki Tesla küresinden renk sensörü ile ham veriler alınarak elde edilen değerler Raspberry Pi cihazı yardımıyla Çizelge 4.1'de ki sayısal renk değerlerine dönüştürülmüştür.

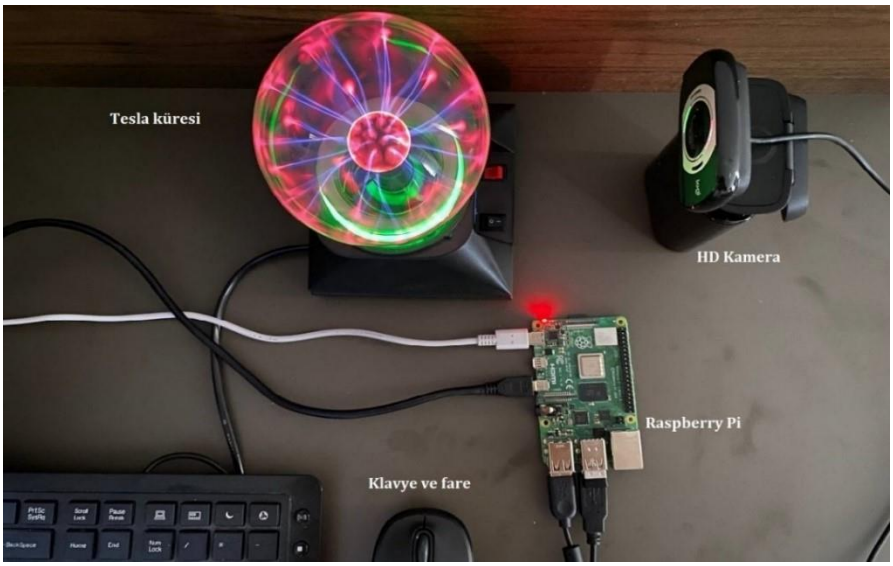
Çizelge 4.1. RGB sensörü ham verileri

Sıra No.	Renk Hexadecimal Kodu	R-G-B Değerleri
1	FFFFFF	(255,255,255)
2	2D2D2D	(45,45,45)
3	FFFF2D	(255,255,45)
4	FFFF2D	(255,255,45)
5	5C5C5C	(92,92,92)
6	5C2020	(92,32,32)
7	5C2020	(92,32,32)
8	5C2020	(92,32,32)

Elde edilen ham verilerden herhangi biri incelendiğinde; RGB değerlerinin (92,92,92), renk sıcaklık değerinin 5201.0K ve renk ışık şiddetinin 44.691 lux olduğu incelenmiştir. Herhangi bir anda elde edilen ham verilerin kaliteli sayılar olduğu ancak zaman geçtikten sonra verilerin birbiriyle ilişkili sonuçlar vererek aynı değerlerin üst üste geldiği gözlemlenmiştir. Renk sensöründen elde edilen (92,32,32) değerlerinin üst üste geldiği, değişken olmadığı ve birbiriyle ilişkili olmasından dolayı rastgele sayı üreteçlerinde tohum değeri olarak kullanılmasının uygun olmayacağı hipotezine ulaşılmıştır. Bu sebeple 'Analiz Sonuçları' başlığı altında, RGB sensöründen elde edilen test sonuçları verilmemiştir.

4.1.2. Webcam Sensöründen Elde Edilen Veriler

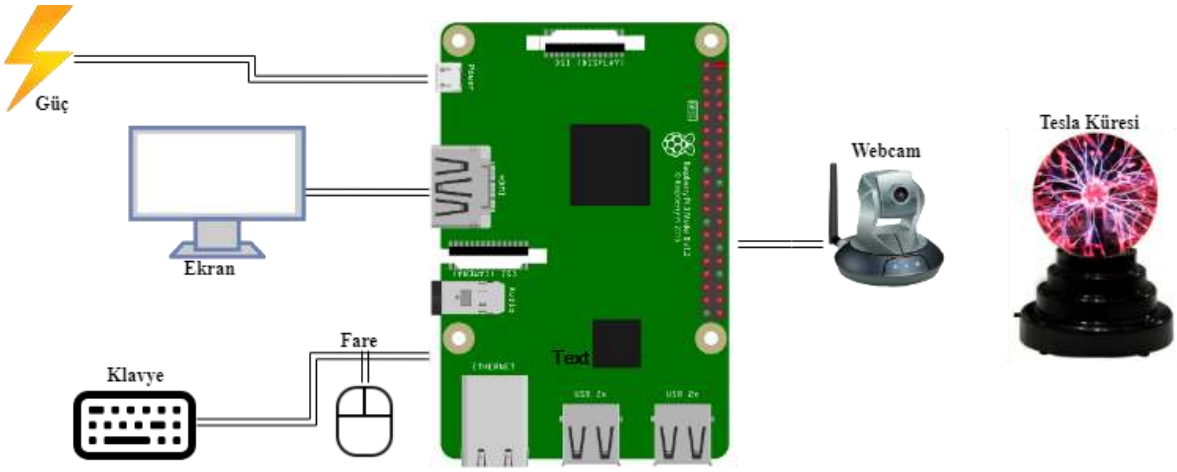
Bu bölümde Raspberry Pi cihazına bağlanan webcam sensörünün entropi kaynağı olarak kullanılan Tesla küresinden elde ettiği ham veriler incelenecektir. Raspberry pi cihazına USB portu ile takılan webcam kullanılarak Tesla küresindeki elektriksel ışımaların hareketleri algılanıp x ve y koordinat değerleri ile ham veriler elde edilmiştir. Webcam kullanılarak tasarlanan deney setinin genel görünüşü Şekil 4.3'te gösterilmiştir.



Şekil 4.3. Webcam sensörlü sistemin genel görünümü

Şematik gösterimi ise Şekil 4.4'de verilen sistem için kullanılan malzemeler aşağıda listelenmiştir:

- Raspberry Pi 4,
- Webcam,
- Tesla Küresi,
- Monitör,
- Klavye ve fare.



Şekil 4.4. Webcam sensörlü deney düzeneğinin şekli

Python programlama dili kütüphanesi olan OpenCV ile küre içerisindeki ışımalar algılanıp, ham veri olarak elde edilen hareketli alanın x ve y koordinatları incelenmiştir. OpenCV açık kaynak kodlu, Intel tarafından 1999 yılında başlatılan görsel bir

kütüphanedir. Akademik projelerde ve ticari ürünlerde kullanılan OpenCV kütüphanesinin gerekli kurulum işlemleri Raspberry pi cihazına yapılmıştır. Kütüphanenin kurulum işlemlerinin bitimiyle, python programlama dili kullanılarak hareketli alanı algılayıp karecik içine alan ve o alanın ağırlık noktasının x ve y koordinatlarını Çizelge 4.2'de ki şekilde veren program geliştirilmiştir.

Çizelge 4.2. Webcam sensörü ham verileri

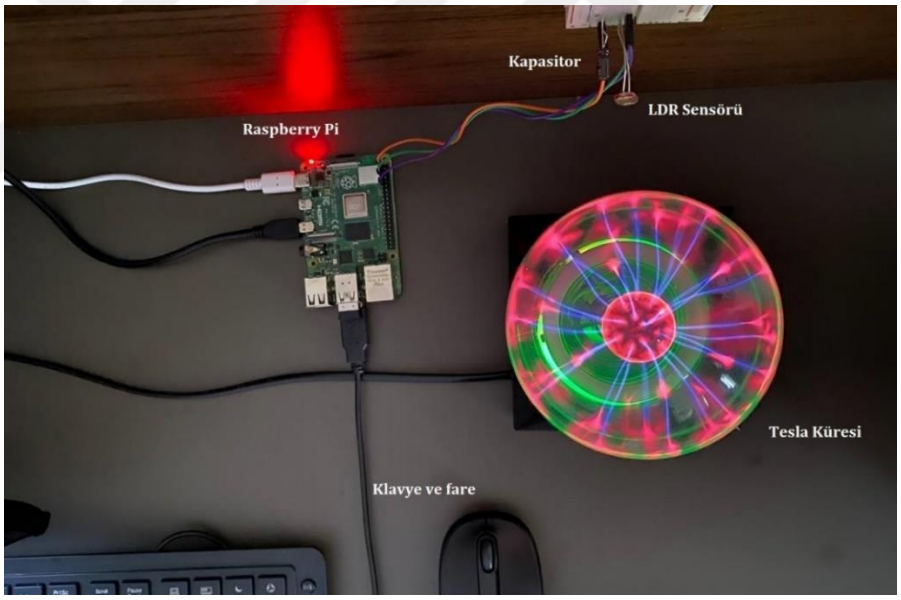
Sıra No.	X Koordinatı	Y Koordinatı	Geçen süre (sn)
1	668	216	0,066
2	277	172	0,085
3	601	333	0,097
4	193	98	0,084
5	158	356	0,089
6	614	477	0,077
7	421	503	0,095
8	385	72	0,073

Yukarıdaki çizelgede elde edilen veriler, rastgele sayı üreteci için tohum değerlerini oluşturmaktadır. Bu değerler kullanılarak 4 farklı yöntem ile sabit uzunlukta çıkışlar elde edilmiştir. Birinci yöntem x ve y koordinat değerlerinin sırasıyla 16 sayısına bölünmesiyle elde edilen kalanın (mod 16) onaltılık sayı sisteminde çevrilmesiyle gerçekleşir. Çizelge 4.2'de dördüncü sıradaki (193 ve 98) değerlerinin mod 16 yöntemiyle işlenmesi sonucu elde edilen kalanlar sırasıyla onaltılık sayı sisteminde '1' ve '2'dir ve bu yöntem ile elde edilen kalan değerlerine göre oluşan sekiz bit uzunluğundaki çıkış (1010 1110)'dır. İstenilen sabit anahtar uzunluğu oluşuncaya kadar hareket algılama döngüsü devam eder. İkinci yöntemde koordinat değerleri ikili sayı sistemine çevrilerek sondaki bitinin alınmasıyla gerçekleşir. Çizelge 4.2'de beşinci sıradaki (158 ve 356) değerlerinin son bitler yöntemiyle ikili sayı sitemine çevrildikten sonra elde edilen son bit değerleri, iki koordinat verisi için de '0' dır. Webcam sensörünün bir hareketi algılamasıyla oluşan x ve y koordinatları bu yöntem ile elde edilen son bit değerlerine göre iki bit uzunluğunda bir çıkış oluşturur. İstenilen sabit anahtar uzunluğu oluşuncaya kadar hareket algılama döngüsü devam eder. Üçüncü ve dördüncü yöntem ise koordinat değerlerinin Md5 ve Sha256 özetleme algoritmalarına sokulmasıyla gerçekleşir. Koordinat değerlerinin (x ve y) ayrı ayrı özeti alınarak XOR son işlemi uygulandıktan sonra çıkış elde edilir. Bu yöntemler sonucunda rastgeleliği kontrol etmek maksadıyla 1.024 bitlik çıkışlar elde edilip monobit rastgelelik testine, 20.000 bitlik çıkışlar elde

edilip ise Poker testine sokulmuştur. Test sonuçları ‘Analiz Sonuçları’ bölümünde değerlendirilecektir.

4.1.3. Işık Sensöründen Elde Edilen Veriler

Bu bölümde Raspberry Pi cihazına bağlanan LDR sensörünün entropi kaynağı olarak kullanılan Tesla küresinden elde ettiği ham veriler incelenecektir. Üzerine düşen ışık miktarına göre ışık şiddetini ölçmeye yarayan, Raspberry Pi cihazında 1µF kapasitör ile çalışan LDR sensörü üzerine düşen ışık miktarına göre kapasitör şarj olacaktır. Raspberry Pi, kapasitörün dolmasını lojik 1 olarak algılayacağından, lojik 1 seviyesine kadar geçen süre ışık şiddetini verecektir. Işık sensörü ve kapasitor, breadboard üzerinden erkek-dişi ara kablolar yardımıyla Raspberry pi cihazında üzerinde bulunan GND, GPIO3 ve 3V3 pinlerine bağlanıp gürültü kaynağı olan Tesla küresinden Çizelge 4.3’de ki gibi ışık değerleri elde edilmiştir. LDR sensörü kullanılarak tasarlanan deney setinin genel görünüşü Şekil 4.5’de gösterilmiştir.

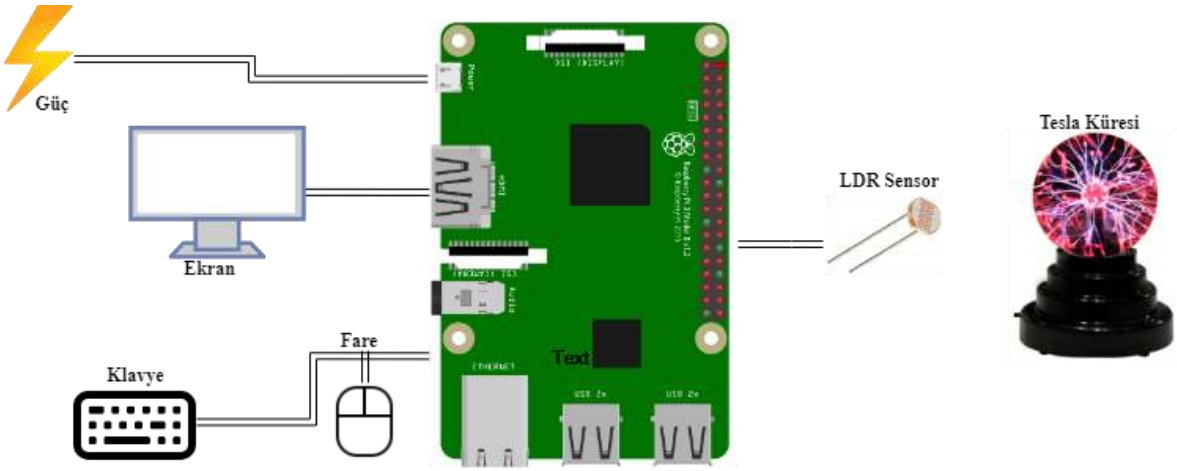


Şekil 4.5. LDR sensörlü sistemin genel görünüşü

Şematik gösterimi ise Şekil 4.6’da verilen sistem için kullanılan malzemeler aşağıda listelenmiştir:

- Raspberry Pi 4,
- LDR Sensörü,
- Tesla Küresi,
- Monitör,

- Klavye ve fare.



Şekil 4.6. LDR sensörlü deney düzeneğinin şekli

Çizelge 4.3. LDR sensörü ham verileri

Sıra No.	Ölçülen ışık şiddeti	Geçen süre (sn)
1	1089	0,1017
2	1083	0,1014
3	1098	0,1021
4	1147	0,1016
5	605	0,1022
6	648	0,1020
7	675	0,1017
8	751	0,1023

Çizelge 4.3'de elde edilen veriler, rastgele sayı üreteci için tohum değerlerini oluşturmaktadır. Elde edilen bu değerler ile; dört farklı yöntem kullanılarak sabit uzunlukta çıkışlar elde edilmiştir. Birinci yöntem ışık değerlerinin 16 sayısına bölünmesiyle elde edilen kalanın (mod 16) onaltılık sayı sistemine çevrilmesi sonucu gerçekleşir. Çizelge 4.3'de ikinci sıradaki (1083) değerinin mod 16 yöntemiyle işlenmesi sonucu elde edilen kalan onaltılık sayı sisteminde 'b'dir ve bu yöntem ile elde edilen kalan değerine göre oluşan dört bit uzunluğundaki çıkış (1011)'dır. İstenilen sabit anahtar uzunluğu oluşuncaya kadar ışık değeri algılama döngüsü devam eder. İkinci yöntem ışık değerinin ikili sayı sistemine çevrilerek sondaki bitinin alınmasıyla gerçekleşir. Çizelge 4.3'de üçüncü sıradaki (1098) değerinin son bitler yöntemiyle ikili sayı sitemine çevrildikten sonra elde edilen son bit değeri '0'dır ve bu yöntem ile elde edilen kalan değerine göre bir bit uzunluğundaki çıkış elde edilir. İstenilen sabit anahtar uzunluğu oluşuncaya kadar ışık değeri algılama döngüsü devam eder. Üçüncü ve dördüncü yöntem

ise ışık değerlerinin Md5 ve Sha26 özetleme algoritmalarına sokulmasıyla elde edilmesidir. Bu yöntemler sonucunda rastgeleliği kontrol etmek maksadıyla Webcam algılayıcı sensörü bölümünde olduğu gibi 1.024 bitlik çıkışlar elde edilip monobit rastgelelik testine, 20.000 bitlik çıkışlar elde edilip ise Poker testine sokulmuştur. Test sonuçları 'Analiz Sonuçları' bölümünde değerlendirilecektir.

4.2. Monobit ve Poker Testi Karşılaştırmaları

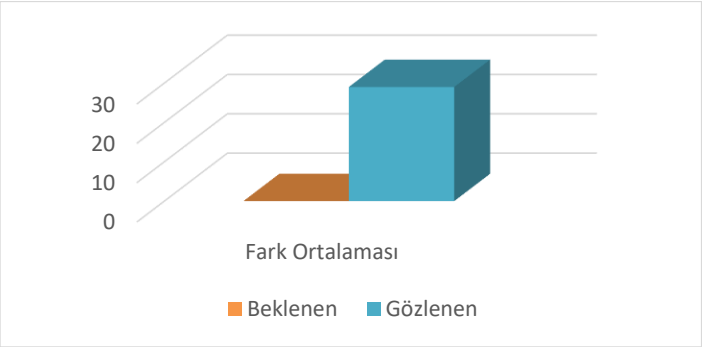
Bu bölümde sözde, webcam ve ışık sensörlerinden elde edilen diziler iki farklı teste tabi tutulmuştur. Her yöntem için 10 adet örnek dizi alınmış, sıfır ve bir sayıları arasındaki fark ortalaması hesaplanarak sonuçlar analiz edilmiştir.

4.2.1. Monobit Testi Sonuçları

Bu çalışmada üç farklı kaynaktan (Sözde, Webcam ve LDR Sensörü) elde edilen değerlerin monobit test sonuçları karşılaştırılmıştır. Frekans testi olarak da bilinen monobit testi, 0 ve 1 sayılarının dizide kaç kez geçtiğinin hesaplanmasıyla bulunur. Önceki bölümde sensörler aracılığıyla elde edilen 1024 bit uzunluğundaki çıkışlarda, 512 bitlik değerin bir, 512 bitlik değerin sıfır olması beklenir. Python programlama dilinde sözde olarak rastgele sayı üreten random fonksiyonu ile elde edilen 1024 uzunluğundaki bit dizisinin monobit test sonucu Çizelge 4.4'de, grafiksel gösterimi ise Şekil 4.7'de verilmiştir. 1.024 bitlik dizi için bir ve sıfır sayıları arasındaki fark ortalamasının 29 olduğu gözlemlenmiştir.

Çizelge 4.4. Sözde rastgele sayı üretiminin monobit testi

S.No.	BEKLENEN	GÖZLENEN		
		0'ların sayısı	1'lerin sayısı	Fark
1	512	525	499	26
2	512	496	528	32
3	512	535	489	46
4	512	516	508	8
5	512	510	514	4
6	512	490	534	44
7	512	493	531	38
8	512	498	526	28
9	512	506	518	12
10	512	538	486	52
Ortalama				**29**

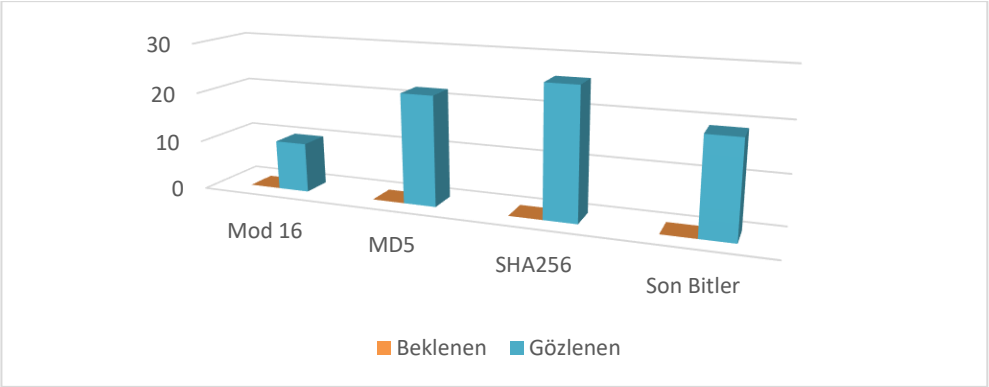


Şekil 4.7. Sözde rastgele sayı üretiminin monobit test grafiği

Tesla küresindeki elektriksel ışımaların Webcam sensörü yardımıyla hareketlerinin algılanıp x ve y koordinat değerleri ile ham veri elde edilerek 4 farklı yöntemle 1.024 bit uzunluğunda oluşturulan çıktı dizilerinin monobit test sonucu Çizelge 4.5'de, grafiksel gösterimi ise Şekil 4.8'de verilmiştir. 1.024 bitlik dizi için bir ve sıfır sayıları arasındaki fark ortalamasının; Mod 16 yöntemi için 10, Md5 yöntemi için 22, Sha256 yöntemi için 26, son bitlerin alınmasıyla oluşturulan dizi için ise 19 olduğu gözlemlenmiştir. Mod 16 yöntemiyle elde edilen dizinin beklenen değerlere en yakın, Sha256 yöntemiyle elde edilen dizinin beklenen değerlere en uzak olduğu görülmüştür. Bu testte Webcam sensörü kullanılarak bütün yöntemlerle üretilen dizideki bir ve sıfır sayıları arasındaki fark ortalamasının, bilgisayarın sözde olarak ürettiği rastgele sayı dizisindeki ortalamaya göre iyi sonuçlar verdiği belirlenmiştir.

Çizelge 4.5. Webcam ile 4 adet yöntemin monobit testi sonuçları

Sayı	BEKLENEN	GÖZLENEN											
		0'ların sayısı				1'lerin sayısı				Fark			
		Mod16	Md5	Sha256	Son Bit	Mod16	Md5	Sha256	Son Bit	Mod16	Md5	Sha256	Son Bit
1	512	521	494	496	494	503	530	528	530	18	36	32	36
2	512	519	497	528	519	505	527	496	505	14	30	32	14
3	512	513	499	516	532	511	525	508	492	2	26	8	40
4	512	499	513	503	505	525	511	521	519	26	2	18	14
5	512	517	508	501	504	507	516	523	520	10	8	22	16
6	512	512	497	527	521	512	527	497	503	0	30	30	18
7	512	510	494	490	503	514	530	534	521	4	36	44	18
8	512	506	492	492	519	518	532	532	505	12	40	40	14
9	512	520	513	496	506	504	511	528	518	16	2	32	12
10	512	511	505	509	509	513	519	515	515	2	14	6	6
Ortalama										**10**	**22**	**26**	**19**

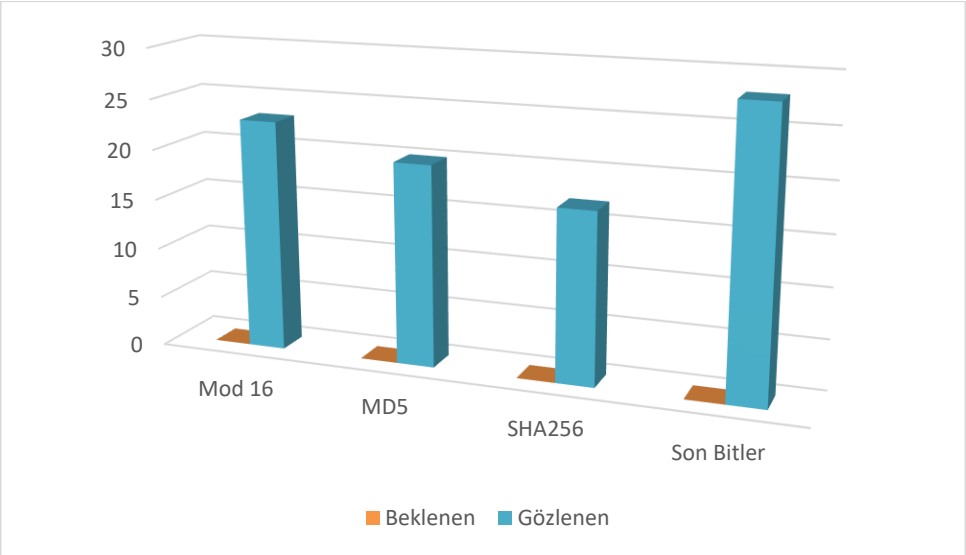


Şekil 4.8. Webcam ile monobit test grafiği

Tesla küresi içindeki ışımaların LDR sensörü ile şiddetini ölçerek 4 farklı yöntemle elde edilen 1.024 bit uzunluğundaki dizilerin monobit test sonucu Çizelge 4.6'da, grafiksel gösterimi ise Şekil 4.9'da verilmiştir. 1.024 bitlik dizi için bir ve sıfır sayıları arasındaki fark ortalamasının; Mod 16 yöntemi için 23, Md5 yöntemi için 20, Sha256 yöntemi için 17, son bitlerin alınmasıyla oluşturulan dizi için 28 olarak gözlemlenmiştir. Sha256 yöntemiyle elde edilen dizinin beklenen değerlere en yakın, son bitler yöntemiyle elde edilen dizinin beklenen değerlere en uzak olduğu görülmüştür. Bu testte LDR sensörü kullanılarak bütün yöntemlerle üretilen dizideki bir ve sıfır sayıları arasındaki fark ortalamasının, bilgisayarın sözde olarak ürettiği rastgele sayı dizisindeki ortalamaya göre iyi sonuçlar verdiği belirlenmiştir.

Çizelge 4.6. LDR sensörü ile 4 adet yöntemin monobit testi sonuçları

Sayı	BEKLENEN	GÖZLENEN								Fark			
		0'ların sayısı				1'lerin sayısı							
		Mod16	Md5	Sha256	Son Bit	Mod16	Md5	Sha256	Son Bit	Mod16	Md5	Sha256	Son Bit
1	512	517	516	504	530	507	508	520	494	10	8	16	36
2	512	505	521	530	522	519	503	494	502	14	18	36	20
3	512	528	516	533	501	496	508	491	523	32	8	42	22
4	512	526	492	519	496	498	532	505	528	28	40	14	32
5	512	495	520	513	521	529	504	511	503	34	16	2	18
6	512	491	532	510	524	533	492	514	500	42	40	4	24
7	512	495	503	519	506	529	521	505	518	34	18	14	12
8	512	514	532	521	533	510	492	503	491	4	40	18	42
9	512	522	514	513	492	502	510	511	532	20	4	2	40
10	512	520	506	524	529	504	518	500	495	16	12	24	34
Ortalama										**23**	**20**	**17**	**28**



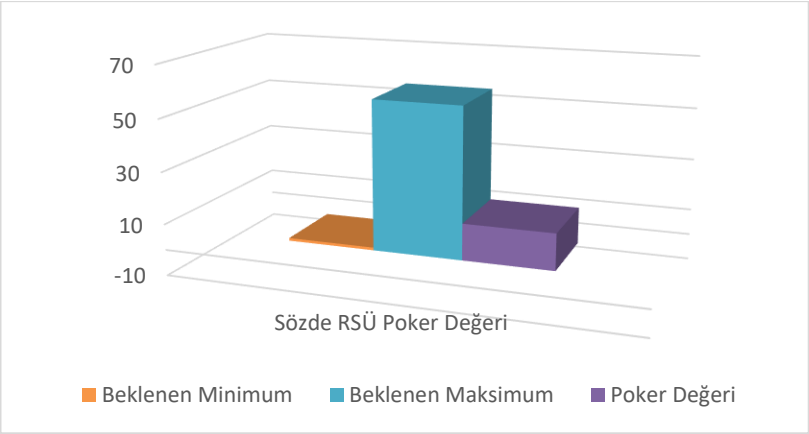
Şekil 4.9. LDR sensörü ile monobit test grafiği

4.2.2. Poker Testi Sonuçları

Bu çalışmada üç farklı kaynaktan (Pseudo, Webcam ve LDR Sensör) elde edilen poker testi sonuçları karşılaştırılmıştır. Bu testte, 20.000 bitlik rastgele bir dizi dörder bitlik bloklara bölünerek 5.000 adet sayı üretilmektedir. Sayılar bu dört bit kullanılarak onaltılık tabanda ifade edilir. Üretilen sayıların rastgele olması ve güvenilirliğini göstermesi için 16 ayrı sayının her birinin eşit sayıda olması beklenir. Üçüncü bölümde belirtilen formül sonucunda hesaplanan poker değerinin testi geçebilmesi için 1,03 ile 57,4 arasında olması gerekmektedir. Python programlama dilinde sözde olarak rastgele sayı üreten Random fonksiyonu ile elde edilen 20.000 uzunluğundaki bit dizisinin poker test sonucu Çizelge 4.7'de, grafiksel gösterimi ise Şekil 4.10'da verilmiştir. 20.000 bitlik dizi için poker değeri 12.7232 olarak gözlemlenmiştir.

Çizelge 4.7. Sözde rastgele sayı üretiminin poker testi

Sayı	BEKLENEN	GÖZLENEN
Poker Değeri (X)	1,03 < X < 57,4	12,7232

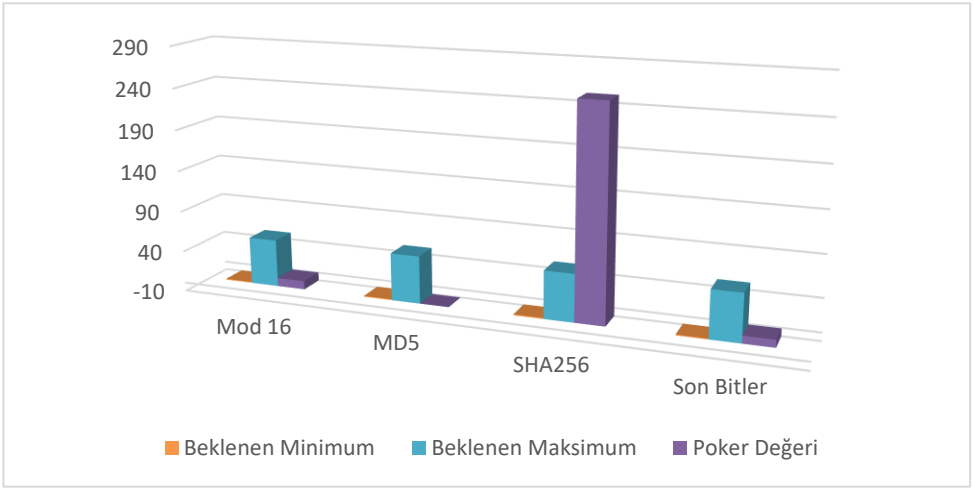


Şekil 4.10. Sözde rastgele sayı üretiminin poker test grafiği

Tesla küresindeki elektriksel ışımaların Webcam sensörü yardımıyla hareketlerinin algılanıp, x ve y koordinat değerleri ile ham veri elde edilerek 4 farklı yöntemle 20.000 bit uzunluğunda oluşturulan çıktı dizilerinin poker test sonucu Çizelge 4.8'de, grafiksel gösterimi ise Şekil 4.11'de verilmiştir. 20.000 bit uzunluğundaki dizi için poker değerinin; Mod 16 yöntemi için 10.1503, Md5 yöntemi için 0.1408, Sha256 yöntemi için 254.2848, son bitlerin alınmasıyla oluşturulan dizi için ise 9.1751 olduğu gözlemlenmiştir. Mod16 ve son bitler yöntemleriyle elde edilen dizilerin bu testten başarıyla geçtiği gözlemlenmiştir.

Çizelge 4.8. Webcam ile poker testi

	BEKLENEN	GÖZLENEN			
		Mod 16	Md5	Sha256	Son Bitler
Poker Değeri (X)	1,03 < X < 57,4	10,1503	0,1408	254,2848	9,1751

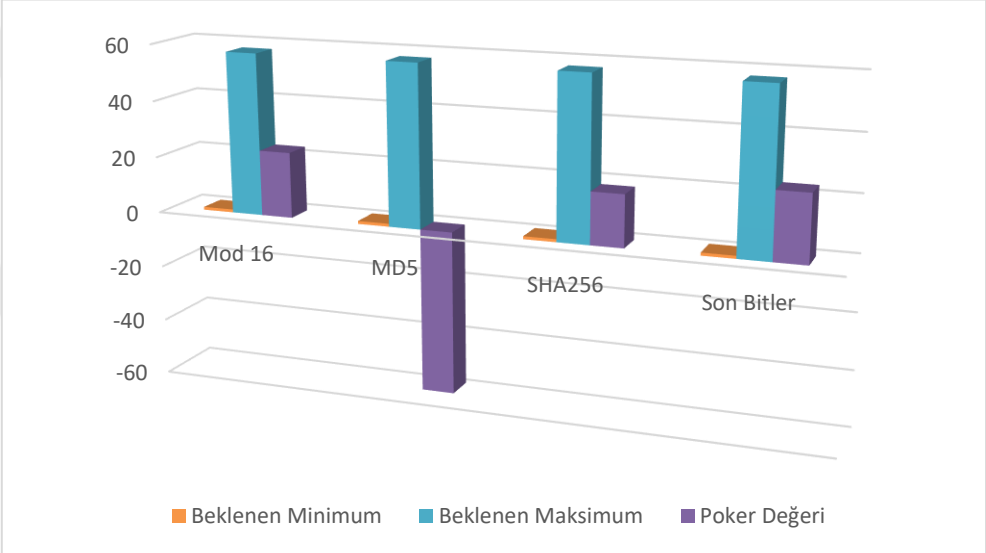


Şekil 4.11. Webcam ile poker test grafiği

Tesla küresi içindeki ışımaların LDR sensörü ile şiddetini ölçerek 4 farklı yöntemle elde edilen 20.000 bit uzunluğundaki dizilerin poker test sonucu Çizelge 4.9'da, grafiksel gösterimi ise Şekil 4.12'de verilmiştir. 20.000 bitlik dizi için poker değerinin; Mod 16 yöntemi için 23.1872, Md5 yöntemi için -57.8751, Sha256 yöntemi için 18.2272, son bitlerin alınmasıyla oluşturulan dizi için 23.5842 olduğu gözlemlenmiştir. Son bitler, mod 16 ve Sha256 yöntemleriyle elde edilen dizilerin bu testten başarıyla geçtiği gözlemlenmiştir.

Çizelge 4.9. LDR sensörü ile poker testi

	BEKLENEN	GÖZLENEN			
		Mod 16	Md5	Sha256	Son Bit
Poker Değeri (X)	1,03 < X < 57,4	23,1872	-57,8751	18,2272	23,5842



Şekil 4.12. LDR sensörü ile poker test grafiği

4.3. FIPS 140-1 Test Sonuçları

Raspberry Pi mini bilgisayarı ile kamera sensörü kullanılarak Tesla küresinden alınan verilerin Mod 16 yöntemiyle son işlem algoritmasına sokularak elde edilen rastgele bitler FIPS 140 testlerine tabi tutulmuştur. Üretilen 20.000 uzunluğundaki bitlerin rastgele olarak kabul görebilmesi için dört adet testten de geçebilmesi gerekmektedir. Bu testler; Monobit, Poker, Run ve Long Run testleridir. Monobit testinde üretilen dizideki birlerin sayısının 9.654 ile 10.346 arasında olması beklenir. Hesaplanan bir sayısının bu aralıkta olması, monobit testinin başarıyla sonuçlandığı anlamına gelmektedir. Poker testinde üretilen dizi için poker değerinin 1,03 ile 57,4 arasında olması beklenir.

Hesaplanan değerin bu aralıkta olması, poker testinin başarıyla sonuçlandığı anlamına gelmektedir. Run testinde üretilen dizide ardı ardına gelen ‘0’ ve ‘1’ rakamlarından oluşan bit blokları sayısının Çizelge 4.10’da belirtilen aralıklar içerisinde olması beklenir. Hesaplanan blok sayılarının bu aralıklarda olması, run testinin başarıyla sonuçlandığı anlamına gelmektedir. Long run testinde ise üretilen dizi için ardışık ‘0’ veya ‘1’ rakamlarının uzunluklarının 34’ten küçük olması beklenir. Hesaplanan ardışık değerlerin bu şartı sağladığı gözlemlenmiştir. 20.000 bit uzunluğunda üretilen 5 farklı testin sonuçları Çizelge 4.10’da belirtilmiştir.

Çizelge 4.10. a) FIPS 140-1 beklenen sonuçlar, b) FIPS 140-1 test sonuçları

Test	**Beklenen**		
Monobit	9654< X <10346		
Poker	1,03< X <57,4		
Run	Blok	Blok Sayısı Aralığı	
	1	2267	2733
	2	1079	1421
	3	502	748
	4	223	402
	5	90	223
	6	90	223
Long Run	<= 34		

a

Test	**1. Test**		**2. Test**		**3. Test**		**4. Test**		**5. Test**	
Monobit	10037		9827		10005		10051		10080	
Poker	7,98		27,30		11,89		19,89		20,85	
Run	(0) Adedi	(1) Adedi	(0) Adedi	(1) Adedi	(0) Adedi	(1) Adedi	(0) Adedi	(1) Adedi	(0) Adedi	(1) Adedi
	2494	2520	2479	2587	2501	2503	2530	2559	2572	2514
	1247	1191	1260	1195	1208	1203	1267	1223	1251	1285
	662	649	616	624	594	616	650	650	592	607
	297	328	313	317	347	319	312	308	303	308
	153	150	164	149	158	161	126	136	164	146
	148	163	179	139	161	166	155	164	147	169
Long Run	Geçti		Geçti		Geçti		Geçti		Geçti	

b

Çizelge 4.10'da görüldüğü gibi 5 ayrı dizi üretilmiştir Üretilen bu dizilerin rastgeleliğini ölçmek maksadıyla 4 adet teste sokulmuştur. Dizilerin tamamının da monobit, poker, run ve long run testlerinden başarılı olarak geçtiği gözlemlenmiştir.

4.4. NIST 800-22 Test Sonuçları

Ulusal düzeyde kabul görmüş olan NIST 800-22, bir rastgele sayı test paketidir. Kendine özgü toplamda 16 ayrı testi olan bu pakette sayının öngörülemezliğini tartışmak için her testin geçilmesi gerekmektedir. Her testin sonucu 'P' olarak adlandırılan bir değere aktarılmalı ve burada uygun şekilde analiz edilmelidir. Her seviyede $P \geq 0.01$ gereksiniminin karşılanması sonucu test başarılı sayılır. Mod 16 yöntemi ve kamera sensörü ile elde edilmiş bir milyon adet verinin test sonuçları Çizelge 4.11'de gösterilmektedir. Bir milyon veri üretilirken aynı koordinat değerlerinin tekrar etmesi ihtimaline karşı python programlama dili sistem saati verisi ile Mod 16 yöntem çıktısı XOR işlemine tabi tutulmuş ve karmaşıklık daha da artırılmıştır.

Çizelge 4.11. NIST 800-22 testi sonuçları

S.No.	Test Adı	P değeri	Sonuç
1	Frekans Testi	0,7634	Başarılı
2	Blok Frekans Testi	0,1559	Başarılı
3	Akış Testi	0,8625	Başarılı
4	Blok İçinde En Uzun Bit Yenilemesi Testi	0,7775	Başarılı
5	İkili Matris Rank Testi	0,4399	Başarılı
6	Ayrık Fourier Dönüşümü Testi	0,6872	Başarılı
7	Örtüşmeyen Şablon Testi	0,7312	Başarılı
8	Örtüşen Şablon Eşleştirme Testi	0,0478	Başarılı
9	Maurer'in Evrensel İstatistik Testi	0,3666	Başarılı
10	Doğrusal Karmaşıklık Testi	0,8760	Başarılı
11	Seri Testi -1	0,8307	Başarılı
12	Seri Testi - 2	0,8601	Başarılı
13	Yaklaşık Entropi Testi	0,8676	Başarılı
14	Kümülatif Toplamlar Testi	0,9532	Başarılı
15	Rastgele Gezinimler Testi (x=+1)	0,3708	Başarılı
16	Rastgele Gezinimler Değişken Testi (x=-1)	0,6782	Başarılı

Diziler FIPS test paketinin yanı sıra dünyada rastgeleliği kontrol etmek için kullanılan en zorlu test paketi olan NIST test paketine karşı da test edilmiştir. Her iki test paketi de Mod 16 yaklaşımı kullanıldığında yüksek kaliteli sonuçlar vermektedir. NIST test paketi 1.000.000 dizilim gerektirirken, FIPS test paketi için 20.000 dizilim gerekmiştir. Üretilen dizilerin bu iki test paketindeki her bir testi etkin bir şekilde geçtiği görülmüştür.

5. SONUÇ VE ÖNERİLER

Rassallık, kriptografik uygulamalarda güvenlik ve gizlilik kapsamında en önemli bileşendir. Bu yüzden birçok haberleşme ortamında kullanılan rastgele sayı üreteçlerinin kalitesi tüm sistemin güvenliğini çok fazla şekilde etkilemektedir. Rastgele sayılar, sözde ve gerçek olarak ya da bu ikisinin birleşimi sonucu hibrit şekilde üretilebilirler. Sözde rastgele sayılar, herhangi bir donanım kullanmadan yazılımsal olarak bilgisayarın kendi içerisinde üretilirken, gerçek rastgele sayılar ise bu çalışmada olduğu gibi kaos ortamının sağlandığı bir doğa olayından üretilmektedir. Üretilen bu sayılar, kalitesinin belirlenmesi maksadıyla bazı istatistiki testlere sokulur. RSÜ'lerin güvenliği gürültü kaynağının entropisi ile yakından ilişkilidir. Tesla küresinin gürültü kaynağı olarak kullanılması ile istenilen kaotik ortamın sağlandığı yapılan analizler ile ispatlanmıştır. Bu çalışmada IoT sensörleri ile fiziksel ortamdan tohum değerinin elde edilmesiyle entropi seviyesinin yükseltilmesi hedeflenmiştir. Raspberry Pi ortamında tasarımı gerçekleştirilen ışık ve webcam sensörlerinden Python programlama dili ile gürültü kaynağı olan Tesla küresinden ham veriler elde edilmiştir. Bu iki sensör kullanılarak mod16, son bitler, Sha256 ve Md5 yöntemleriyle toplamda sekiz farklı değer elde edilip analizi yapılmıştır. İlk aşamada kullanılan renk sensörü ile elde edilen ham veriler analiz edilmiş, bu ham verilerin art arda gelerek deterministik bir yapıya sahip olduğu tespit edilmiş ve istenilen kaotik durumu sağlamayacağı hipotezine ulaşılmıştır. Monobit ve Poker testi ile yapılan analizler sonucunda, Mod 16 yöntemiyle ve kamera sensörüyle alınan ham verilerin diğer yöntemlere göre kaliteli sayılar ürettiği ve iyi sonuçlar verdiği gözlemlenmiştir. Mod 16 yöntemiyle elde edilen diziler, rastgeleliğin ölçülmesi maksadıyla FIPS 140-1 ve NIST 800-22 test paketlerine sokulmuştur. Tasarlanan projede elde edilen diziler, bu iki test paketini de başarıyla geçmiş böylece rastgelelikleri teyit edilmiştir.

Sonuç olarak bu projeyle, günümüzde hızla gelişen ve ileride günlük yaşantımızda daha fazla karşılaşacağımız IoT sistemleri sensörlerinden rastgele sayı üretimi için tohum değerlerinin elde edilebileceği gözlemlenmiştir. Çalışmayı diğerlerinden özgün kılan yönü Tesla küresinin gürültü kaynağı olarak seçilmesi ile tekrar etmeyen, bir sonraki adımda tahmin edilemeyen verilerin elde edildiği kaos ortamının oluşturulmasıdır. Önceki çalışmalarda sistemin otonom bir şekilde çalışmadığı, dışarıdan insan müdahalesi ile rastgele dizilerin elde edildiği incelenmiştir. Bu çalışmada ise dışarıdan herhangi bir insan müdahalesi gerektirmeden otonom bir yapıda dizilerin üretildiği ve test maksatlı olarak mobil uygulamada ham verilerin izlendiği görülmektedir. Uzaktan izleme

sistemini, firebase veritabanı ve MIT app inventor mobil uygulama geliştirme ortamı oluşturmaktadır. Android mobil uygulaması ile webcam sensöründen elde edilen x ve y koordinatları anlık olarak izlenebilmektedir. Küre içindeki merkezden çıkan rastgele elektriksel ışımalardan elde edilen anahtarların sözde rastgele sayı üreteçlerinden elde edilen anahtarlara göre iyi sonuçlar verebildiği kanıtlanmıştır.

KAYNAKLAR

Abood, O. G., Guirguis, S. ve Guirguis, S. K. (2018). A survey on cryptography algorithms. *International Journal of Scientific and Research Publications*, *8*(7), 495–516.

Akkoca, E. ve Buluş, E. (2021). Telea ve naiver stokes algoritmaları kullanılarak görüntülerdeki bozulmaları düzeltme. *TBV Journal of Computer Science and Engineering*, *14*(2), 77–85.

Al Salami, S., Baek, J., Salah, K. ve Damiani, E. (2016). Lightweight encryption for smart home. *2016 11th International Conference on Availability, Reliability and Security, ARES*, 382–388.

Ali, A. M. ve Farhan, A. K. (2020). A novel improvement with an effective expansion to enhance the MD5 hash function for verification of a secure e-document. *IEEE Access*, *8*, 80290–80304.

Ansari, U., Chaudhary, A. K. ve Verma, S. (2022). True random number generator (TRNG) using sensors for low cost IoT applications. *2022 International Conference on Communication, Computing and Internet of Things, IC3IoT 2022*, 1-6.

Arroyo, J. T. C. ve Delima, A. J. P. (2020). An improved affine cipher using blum blum shub algorithm. *Article in International Journal of Advanced Trends in Computer Science and Engineering*, *9*(3), 3295–3298.

Atar, E., Ersoy, O. K. ve Özyılmaz, L. (2017). Dik eşleştirme arayış yöntemi ile hibrit veri sıkıştırma ve optiksel kriptografi. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, *32*(1), 139–147.

Bakır, D. (2017). *Rastgele Sayı Üreteçlerinin Donanımsal Olarak Gerçekleştirilmesi*. (Yüksek Lisans Tezi). Munzur Üniversitesi Fen Bilimleri Enstitüsü, Tunceli, Türkiye.

Chen, I. Te. (2013). Random numbers generated from audio and video sources. *Mathematical Problems in Engineering*.

Coşkun, A. ve Ülker, Ü. (2013). Ulusal bilgi güvenliğine yönelik bir kriptografi algoritması geliştirilmesi ve harf frekans analizinine karşı güvenirlik tespiti. *Journal of Information Technologies*, *6*(2), 31.

Dash, S. ve Choudekar, P. (2021). Home automation using smart devices and IoT. *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions), ICRITO*, 1–5.

Demirkol, A. Ş. (2007). *Kaotik Osilatör Girişli Adc Tabanlı Rastgele Sayı Üreteci*. (Yüksek Lisans Tezi). İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, Türkiye.

Diem, C. (2016). Cryptology: Methods, applications and challenges. *Mathematics and Society*, 221–250.

Etem, T. ve Kaya, T. (2020). Görüntü şifreleme için trivium-doğrusal eşlenik üreteci tabanlı bit üretimi. *Fırat University Journal of Engineering Science*, *32*(1), 287–294.

Genç, Y. ve Arslan Tuncer, S. (2019). İnsan hareketleri tabanlı gerçek rasgele sayı üretimi. *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*, *8*(1), 261–269.

Gowda, S. N. (2016). Innovative enhancement of the caesar cipher algorithm for cryptography. *2016 2nd International Conference on Advances in Computing, Communication, & Automation (ICACCA)*, 1–4.

Grime, J. (2013). Maths from the talk "Alan Turing and the Enigma Machine". *Singingbanana.* Erişim adresi: http://www.singingbanana.com/enigmaproject/maths.pdf.

Huang, M., Chen, Z., Zhang, Y. ve Guo, H. (2020). A phase fluctuation based practical quantum random number generator scheme with delay-free structure. *Applied Sciences*, *10*(7), 2431.

Kasgar, A. K., Agrawal, J. ve Shahu, S. (2012). New modified 256-bit md 5 algorithm with sha compression function. *International Journal of Computer Applications*, *42*(12).

Katyal, R., Mishra, A. ve Baluni, A. (2013). True random number generator using fish tank image. *International Journal of Computer Applications*, *78*(16), 0975–8887.

Khawas, C. ve Shah, P. (2018). Application of firebase in android app development-a study. *Article in International Journal of Computer Applications*, *179*(46), 49–53.

Kour, V. P. ve Arora, S. (2020). Recent developments of the internet of things in agriculture: a survey. *IEEE Access*, *8*, 129924–129957.

Le, C., Grande, A. M., Carmine, A. J., Thompson, J. ve Khan Mohd, T. (2022). Analysis of various vulnerabilities in the raspbian operating system and solutions. *2022 IEEE World AI IoT Congress, AIIoT*, 659–664.

Li, W. J., Yen, C., Lin, Y. S., Tung, S. C. ve Huang, S. M. (2018). Just IoT internet of things based on the firebase real-time database. *Proceedings - 2018 IEEE International Conference on Smart Manufacturing, Industrial and Logistics Engineering, SMILE.* Hsinchu, Taiwan: Institute of Electrical and Electronics Engineers Inc.

Luengo, E. A., Cerna, M. B. L., Villalba, L. J. G., Hurley-Smith, D. ve Hernandez-Castro, J. (2022). Critical analysis of hypothesis tests in federal information processing standard (140-2). *Entropy 2022, 24*(5), 613.

Maksimović, M., Vujović, V., Davidović, N., Milošević, V. ve Perišić, B. (2014). Raspberry pi as internet of things hardware: performances and constraints. *Design*

Issues, *3*(8), 1-6.

Mantoro, T. ve Lazuardi, Y. (2017). SMS based home appliance security approach using ROT 13, RC4 and RSA algorithm. *3rd International Conference on Computing, Engineering, and Design, ICCED 2017* (1–5). Kuala Lumpur, Malaysia: IEEE.

Maqsood, F., Ahmed, M., Mumtaz Ali, M. ve Ali Shah, M. (2017). Cryptography: a comparative analysis for modern techniques. *IJACSA) International Journal of Advanced Computer Science and Applications*, *8*(6).

Marot, J. ve Bourennane, S. (2017). Raspberry pi for image processing education. *25th European Signal Processing Conference, EUSIPCO 2017* (2364–2368). Kos, Greece: IEEE.

Mezaal, Y. S. ve Abdulkareem, S. F. (2017). Affine cipher cryptanalysis using genetic algorithms. *JP Journal of Algebra, Number Theory and Applications*, *39*(5), 785–802.

Mir, S. B. ve Llueca, G. F. (2020). Introduction to programming using mobile phones and mit app inventor. *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje*, *15*(3), 192–201.

Morkel, T. ve Eloff, J. (2004). Encryption techniques: A timeline approach. *Information and Computer Security Architecture (ICSA) Research Group*.

Nagpal, G. S., Singh, G., Singh, J. ve Yadav, N. (2018). Facial detection and recognition using opencv on raspberry pi zero. *IEEE 2018 International Conference on Advances in Computing, Communication Control and Networking, ICACCCN* (945–950). Greater Noida, India: IEEE.

Naru, E. R., Saini, H. ve Sharma, M. (2017). A recent review on lightweight cryptography in IoT. *2017 International Conference on IoT in Social, Mobile, Analytics and Cloud, I-SMAC* (887–890). Palladam, India: IEEE.

Neha, S. (2017). A review of information security using cryptography technique. *International Journal of Advanced Research in Computer Science*, *8*(4).

Ozkaynak, F., Ozdemir, H. I. ve Ozer, A. B. (2015). Cryptographic random number generator for mobile devices. *2015 23rd Signal Processing and Communications Applications Conference*, 1733–1736.

Panuganti, S. L., Gajula, N. H., Rathnala, P., Patnaik, M. P. K. ve Sura, S. R. (2022). Embedded Color Segregation using Arduino. *2022 Sixth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), I-SMAC*, 832–838.

Park, B. K., Park, H., Kim, Y. S., Kang, J. S., Yeom, Y., Ye, C., … Han, S. W. (2019). Practical true random number generator using cmos ımage sensor dark noise. *IEEE Access*, *7*, 91407–91413.

Prasad, K. ve Kumari, M. (2020). A review on mathematical strength and analysis of

Enigma. *arXiv preprint arXiv:2004.09982*.

Qutieshat, A., Aouididi, R. ve Arfaoui, R. (2019). Design and construction of a low-cost arduino-based ph sensor for the visually impaired using universal pH paper. *Journal of Chemical Education*, *96*(10), 2333–2338.

Rachmawati, D., Tarigan, J. T. ve Ginting, A. B. C. (2018). A comparative study of message migest 5(md5) and sha256 algorithm. *Journal of Physics: Conference Series* (*978*(1), 012116). Medan, Indonesia: IOP Publishing.

Rehman, A. U., Hussain, M., Munawar, A., Attique, M., Idress, M., Anwar, F. ve Ahmad, M. (2020). E-cultivation using the IoT with adafruit cloud systematic review of real-time remote health monitoring system in triage and priority-based sensor technology: taxonomy, open challenges, motivation and recommendations view project health information exchan. *Article in International Journal of Advanced And Applied Sciences*, *7*(9), 75–82.

Sainz-Raso, J., Martin, S., Diaz, G. ve Castro, M. (2019). Security vulnerabilities in raspberry pi-analysis of the system weaknesses. *IEEE Consumer Electronics Magazine*, *8*(6), 47–52.

Salim, G. M., Ismail, H., Debnath, N. ve Nadya, A. (2015). Optimal light power consumption using LDR sensor. *2015 IEEE international symposium on robotics and intelligent sensors (IRIS)*, 144–148.

Saracel, N. ve Aksoy, I. (2020). Toplum 5.0: Süper akıllı toplum. *Social Sciences Research Journal*, *9*(2), 26–34.

Sezgin, Z. E. (2021). Tesla bobini. *Docplayer.* Erişim adresi: https://docplayer.biz.tr/217850236-Tesla-bobini-zafer-emre-sezgin.html adresinden erişildi.

Sharma, A., Khan, F., Sharma, D., Gupta, S. ve Student, F. Y. (2020). Python: The programming language of future. *International Journal of Innovative Research in Technology*, *6*(12), 115–118.

Sharma, M. ve Garg, R. B. (2016). DES: The oldest symmetric block key encryption algorithm. *Proceedings of the 5th International Conference on System Modeling and Advancement in Research Trends, SMART* (53–58). Moradabad, India: IEEE.

Srinath, K. R. (2017). Python-the fastest growing programming language. *International Research Journal of Engineering and Technology*, *4*(12), 354–357.

Stipčević, M. ve Koç, Ç. K. (2014). True random number generators. *Open Problems in Mathematics and Computational Science*, 275–315.

Sunny, A. I., Zhao, A., Li, L. ve Sakiliba, S. K. (2020). Low-cost ıot-based sensor system: a case study on harsh environmental monitoring. *Sensors, 21*(1), 214.

Tavas, V. (2011). *Tümleştirmeye Uygun Rastgele Sayı Üreteçleri*. (Doktora Tezi).

İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, Türkiye.

Üçgün, H., Gömbeci, F., Yüzgeç, U. ve Yalçin, N. (2020). IoT tabanlı platform ile gerçek zamanlı iç ortam hava kalitesi izleme sistemi. *BSEU Journal of Science Araştırma Makalesi-Research Article*, *7*(1), 370–381.

Wibowo, P. A. P. ve Romadhon, F. W. (2017). Generation of pseudorandom numbers from audio input in smart phone Android. *Proceeding of 2017 11th International Conference on Telecommunication Systems Services and Applications* (1–5). Lombok, Indonesia: IEEE.

Xie, B. ve Abelson, H. (2016). Skill progression in MIT app inventor. *2016 IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC*, 213–217.

Yadav, G. S. ve Ojha, A. (2013). A novel visual cryptography scheme based on substitution cipher. *2013 IEEE 2nd International Conference on Image Information Processing, IEEE ICIIP*, 640–643.

Yaşar, S. N., Ceren Dikici, F., Tanyildizi, E. ve Karaköse, E. (2021). Fen ve mühendislik çalışmalarındaki rastgelelik gereksinimleri için orta kare ve sha3 algoritmasını temel alan bir üreteç tasarımı. *Fırat Üniversitesi Fen Bilimleri Dergisi*, *33*(1), 81–91.

Yılmaz, M. ve Ballı, S. (2016). Veri şifreleme algoritmalarının kullanımı için akıllı bir seçim sistemi geliştirilmesi. *Uluslararası Bilgi Güvenliği Mühendisliği Dergisi*, *2*(2), 18–28.

Yosunlu, D. ve Avaroğlu, E. (2020). Son işlem algoritmalarının incelenmesi. *Bilgisayar Bilimleri ve Teknolojileri Dergisi*, *1*(2), 66–73.

Zhang, X., Qi, L., Tang, Z. ve Zhang, Y. (2014). Portable true random number generator for personal encryption application based on smartphone camera. *Electronics Letters*, *50*(24), 1841–1843.