FIRAT ÜNİVERSİTESİ

TEKNOLOJİ FAKÜLTESİ
Yazılım Mühendisliği Bölümü

YMH321 – Bilgi Sistemleri Ve Güvenliği
Dersi Uygulaması ve Dokümantasyonu

Fiziksel Entropi İle Gerçek Rastgele Sayı Üretimi

Geliştiren
245541023 Mehmet Onur Boyraz

1. GİRİŞ VE PROJE AMACI

Bilgisayar bilimlerinde kullanılan standart `random()` fonksiyonları, belirli matematiksel formüllere dayandığı için deterministiktir ve tamamen tahmin edilemez değildir (Pseudo-Random)[1]. Bu projenin temel amacı, bu kısıtlamayı aşarak fiziksel dünyanın öngörülemezliğinden (entropi) beslenen bir **"Gerçek Rastgele Sayı Üreteci" (True Random Number Generator - TRNG)** geliştirmektir[2].

2. SİSTEM MİMARİSİ VE ÇALIŞMA MANTIĞI

Sistem, fiziksel dünyadan alınan veriyi matematiksel kaos ile harmanlayan hibrit bir mimariye sahiptir[3].

1. **Fiziksel Entropi Kaynağı (Kamera):** Bilgisayarın kamerasından anlık görüntü alınarak ortamdaki ışık değişimleri ve sensör üzerindeki termal gürültü (noise) toplanır[4444].
2. **Kriptografik Özetleme (SHA-256):** Alınan ham görüntü verisi, endüstri standardı SHA-256 algoritması ile hashlenerek benzersiz bir sayısal "tohum" (seed) değerine dönüştürülür[5555].

Matematiksel Kaos (Collatz Sanısı): Elde edilen tohum, matematikteki ünlü $3x+1$ problemi (Collatz Sanısı) algoritmasına sokularak karıştırılır ve "kaotik" bir yapıya büründürülür[6].

2.1. GÖRÜNTÜ VERİSİNİN İŞLENMESİ VE ENTROPİ ÇIKARIMI

Sistemin rastgelelik kaynağı (entropi havuzu), bilgisayar kamerası tarafından yakalanan fotonlar ve görüntü sensöründeki (CMOS/CCD) termal gürültüdür. Bu fiziksel verinin sayısal bir "tohum" (seed) değerine dönüştürülmesi üç aşamada gerçekleştirilir:

1. **Ham Veri Yakalama (Raw Data Acquisition):** OpenCV kütüphanesi kullanılarak kameradan anlık bir kare (frame) yakalanır. Bu kare, matematiksel olarak her biri 0-255 arasında değişen piksel değerlerinden (RGB) oluşan devasa bir matristir. Ortamdaki en ufak ışık değişimi veya sensördeki elektronik dalgalanma, bu matrisin değerlerini mikroskobik düzeyde değiştirir.
2. **Byte Dizisine Dönüştürme (Serialization):** Matris formundaki bu görüntü verisi, işlenebilir tek boyutlu bir veri akışına dönüştürülmelidir. Bunun için görüntü matrisi, Python'daki `tobytes()` fonksiyonu ile ham bir **Byte Dizisine (Byte Stream)** çevrilir. Bu işlem, görüntüdeki renk, parlaklık ve gürültü bilgisini içeren milyonlarca bitlik bir veri bloğu oluşturur.

3. **Kriptografik Özetleme (Hashing - SHA-256):** Elde edilen ham byte dizisi, **SHA-256 (Secure Hash Algorithm)** fonksiyonuna girdi olarak verilir.
    - **Neden SHA-256?** Görüntüdeki milyonlarca pikselden sadece bir tanesinin değeri bile değişse (örneğin; havada uçuşan bir toz zerresi veya sensördeki 1 bitlik ısınma hatası), SHA-256 algoritması **Çığ Etkisi (Avalanche Effect)** prensibi gereği tamamen farklı bir çıktı üretir.
    - **Sonuç:** Bu işlem sonucunda, fiziksel dünyanın kaosu, 256 bitlik (64 karakterlik hexadecimal) benzersiz ve tahmin edilemez bir tam sayıya (Integer Seed) dönüştürülmüş olur.

Özet Akış Şeması: `Fotonlar -> Sensör Matrisi (Piksel) -> Byte Dizisi -> SHA-256 Algoritması -> Rastgele Sayı Tohumu (Seed)`

3. PROBLEM TESPİTİ: İSTATİSTİKSEL YANLILIK (BIAS)

Sistemin ilk prototipi üzerinde yapılan **"Mini Turing Testleri"** sonucunda ciddi bir istatistiksel dengesizlik (Bias) tespit edilmiştir[7].

- **Gözlemlenen Sorun:** Kamera önüne engel geldiğinde, ortam çok karanlık veya aşırı parlak olduğunda sensör verileri tek düzeleşmektedir[8].
- **Test Verileri:** İlk testlerde üretilen sayılarda çift sayıların baskın olduğu görülmüştür:
    - **Çift Sayılar:** 67 adet (%67)
    - **Tek Sayılar:** 33 adet (%33)[9].
- **Sonuç:** %67'lik bu yığılma, sistemin fiziksel koşullardan aşırı etkilendiğini ve kriptografik güvenilirliğinin düşük olduğunu göstermiştir[10].

4. ÇÖZÜM: VON NEUMANN AYRIŞTIRICISI (EXTRACTOR)

Fiziksel kaynaktan gelen bu yanlılığı gidermek amacıyla sisteme **Von Neumann Ayrıştırıcısı** entegre edilmiştir[11].

- **Çalışma Prensibi:** Algoritma gelen bitleri (0 ve 1) ikişerli gruplar halinde işler:
    - `01` gelirse -> `0` olarak kabul edilir.
    - `10` gelirse -> `1` olarak kabul edilir.
    - `00` veya `11` gelirse -> **Veri çöpe atılır (Discard)**[12].
- **Fail-Safe Mekanizması:** Bu yöntem sadece dengeyi sağlamakla kalmaz, aynı zamanda bir güvenlik sigortasıdır. Kamera tamamen kapatılırsa (siyah ekran), sistem sadece `00` üreteceğinden algoritma çıktı vermeyi durdurur. Hatalı veri üretmektense hiç üretmemek tercih edilmiştir[13131313].

5. DOĞRULAMA VE TEST SONUÇLARI

Von Neumann entegrasyonu sonrası sistem, zorlu koşullar altında (kamera önünde engel varken) tekrar test edilmiştir[14].

- **Sayısal Denge Sonucu:**
  - **Çift Sayılar:** 52 adet (%52.00)
  - **Tek Sayılar:** 48 adet (%48.00)[15].
  - *Değerlendirme:* Önceki %67'lik sapma hatası tamamen giderilmiş, ideal dengeye ulaşılmıştır[16].
- **Mühendislik Ödünleşimi (Trade-off):** Bit bazında hafif sapmalar (%55-%45) gözlemlenmiştir. Bunun nedeni, düşük entropili kaynaklarda (karanlık ortam) Von Neumann algoritmasının verinin %75-80'ini çöpe atması ve örneklem kümesini küçültmesidir. Bu durum, veri kalitesi (rastgelelik) için hızdan feragat edilen bilinçli bir mühendislik tercihidir[17171717].

6. GÜVENLİK ANALİZİ

Sistemin siber güvenlik açısından değerlendirmesi aşağıdadır[18]:

Katman	Güvenlik Puanı	Gerekçe
Kaynak (Kamera)	⚠ Orta/Düşük	Fiziksel dünya dışarıdan manipüle edilebilir ama yinede sıradan bir random sayı üretecinden milyon kat daha güvenlidir.
İşleme (SHA-256)	🛡 Çok Yüksek	Endüstri standardı şifreleme, geri döndürülemez.
Karıştırma (Collatz)	🎲 Orta	Güvenlikten ziyade sisteme "kaotik karmaşıklık" katar.
Temizlik (Von Neumann)	✅ Yüksek	İstatistiksel yanlılığı (bias) matematiksel kesinlikle yok eder.
GENEL PUAN	**7.5 / 10**	*(Profesyonel/Ticari sistemler 9.5+ olmalıdır)*

Savunma Senaryoları:

1. **Siyah Bant Saldırısı:** Saldırgan kamerayı kapatırsa, Von Neumann `00` verilerini çöpe atarak sistemin "kilitlenmesini" sağlar, böylece tahmin edilebilir veri üretimi engellenir[23].

2. **Man-in-the-Middle:** Zararlı yazılım kamerayı taklit ederse, ardışık hash kontrolü ile sistemin donduğu tespit edilebilir[24].

7. TEKNİK EKLER: ALGORİTMA VE AKIŞ

Algoritma Sözde Kodu (Pseudo-Code):

Plaintext

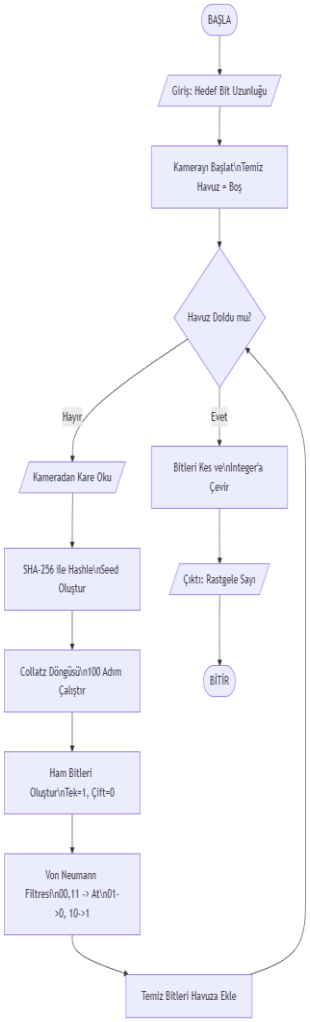
```
BAŞLA
    FONKSİYON Von_Neumann_Temizle(bit_dizisi):
        DÖNGÜ (bitleri ikişerli al):
            EĞER "01" -> "0" Ekle
            EĞER "10" -> "1" Ekle
            EĞER "00" veya "11" -> Atla
        DÖNDÜR Temiz_Bitler

    ANA DÖNGÜ (Yeterli bit toplanana kadar):
        1. Görüntü Yakala (Kamera)
        2. Hashle (SHA-256) -> Seed
        3. Collatz Karıştırma (Seed -> Ham Bitler)
        4. Von Neumann ile Temizle

    Bitleri Sayıya Çevir ve Yazdır
BİTİR [cite: 96-133]
```

Akış Şeması Özeti:

Süreç BAŞLA komutuyla kamerayı açar, Temiz_Bit_Havuzu dolana kadar döngüye girer. Her döngüde görüntü hashlenir, Collatz ile karıştırılır ve Von Neumann filtresinden geçirilir. Yeterli bit toplandığında tam sayıya çevrilerek sonuç üretilir 25.



SONUÇ: Bu proje, tek kaynaklı fiziksel sistemlerin en büyük sorunu olan "bias" problemini Von Neumann mimarisi ile çözen, güvenli ve gerçekçi bir TRNG prototipidir[26].