# KRİPTOGRAFİK GÜVENLİ RASTGELE SAYI ÜRETECİ (CSPRNG)

## 1. Giriş

Python programlama dili ile geliştirilen ve kriptografik standartlara uygun olarak tasarlanan bir **Sözde Rastgele Sayı Üreteci (PRNG)** algoritmasını oluşturduk. Algoritma, basit lineer üreteçlerin aksine, saldırganlar tarafından tahmin edilemezlik ve kolay olarak kırılmamazlık sağlamak üzere optimize edilmiştir.

## 2. Algoritmik Yapı ve Metodoloji

Sistem, temel olarak üç ana bileşen üzerinden çalışmaktadır:

### 2.1. Entropi Kaynağı (Seeding)

Algoritmanın kalbi olan os.urandom(32) fonksiyonu, işletim sisteminin donanımsal kesmelerinden ve çevresel gürültülerden topladığı gerçek rastgele veriyi (32 byte) kullanır. Bu, algoritmanın başlangıç noktasının (seed) dışarıdan tahmin edilmesini engeller.

### 2.2. Durum Güncelleme ve Hash Chaining

Algoritma, çıktı üretirken **SHA-256** kullanır.

- Her üretim aşamasında mevcut "state" (durum), bir sonraki durumun girdisi olur.
- SHA-256'nın tek yönlü fonksiyon (one-way function) özelliği sayesinde, üretilen bir sayıdan bir önceki sayıya veya ana "seed" değerine geri dönülmez.

### 2.3. Dinamik Yeniden Tohumlama (Reseed)

Sürekli aynı havuzdan sayı üretmek uzun vadede desenlerin oluşmasına neden olabilir. Bu algoritma, her 1000 üretimde bir (reseed_counter > 1000) sistemi yeni bir sistem entropisi ve nano saniye hassasiyetinde zaman damgası (time.time_ns()) ile günceller.

## 3. İstatistiksel Analiz ve Güvenlik Parametreleri

| Parametre | Analiz Sonucu |
|---|---|
| **Algoritma Türü** | Hash-Based CSPRNG (SHA-256) |
| **Güvenlik Seviyesi** | 256-bit Güvenlik (Kriptografik Seviye) |
| **Periyot** | Teorik olarak sonsuza yakın ($2^{256}$ durum) |
| **Dağılım Tipi** | Üniform (Eş Dağılımlı) |

### 3.1. Üniform Dağılım Testi

Algoritmanın get_int(min, max) fonksiyonu, modülo aritmetiği kullanarak sayıları istenen aralığa indirger. Büyük örneklemlerde (örneğin 100.000 üretim), her bir sayının seçilme olasılığı eşit dağılım göstermekteddir.

## 4. Sonuç

Geliştirilen SecureRNG algoritması; tahmin edilemezlik, geriye dönük güvenlik ve yüksek entropi kriterlerini tam olarak karşılamaktadır. Hem akademik analizler hem de pratik uygulamalar (simülasyonlar, şifreleme anahtarı üretimi vb.) için güvenli bir temel sunabilir.

### 4.1. Ekran Çıktısı

```
[aligullu@Ali-MacBook-Pro random sayi algortima % python3 main.py
 Rastgele Sayı (1-100): 79
 Rastgele Hex (32 char): 8b817b7bf8a2b5d6f71c7a6fcd17d9f8
[aligullu@Ali-MacBook-Pro random sayi algortima % python3 main.py
 Rastgele Sayı (1-100): 90
 Rastgele Hex (32 char): 881ef1ff5dfedfb6d6e1b3b487284f52
[aligullu@Ali-MacBook-Pro random sayi algortima % python3 main.py
 Rastgele Sayı (1-100): 13
 Rastgele Hex (32 char): f22c71cc063588d5aeba7834f8ce5538
```