PROJE DOKÜMANTASYONU - SecureRNG
Kriptografik Güvenli Rastgele Sayı Üreteci

1.PROJE TANIMI

Bu proje, kullanıcı seed değeri, zaman damgası ve sistem rastgeleliğini birleştirerek SHA-256 hash algoritması ile kriptografik güvenli rastgele sayılar üreten bir Python modülüdür.

Dosya Adı: secure_random_generator.py
Programlama Dili: Python
Toplam Satır: 128
Amaç: Kriptografik güvenli rastgele sayı üreteci

2. KULLANILAN TEKNOLOJİLER VE KÜTÜPHANELER

hashlib: SHA-256 hash algoritması için kullanılır
time: Nanosaniye hassasiyetinde zaman damgası almak için kullanılır
os: İşletim sistemi rastgeleliği (os.urandom) için kullanılır

3. SINIF YAPISI: SecureRandomGenerator

3.1. Constructor Metodu (**init**)

Parametre: user_seed (4 basamaklı sayı, 1000-9999 arası)
İşlev: Üreteci başlatır ve başlangıç durumunu oluşturur
Validasyon: Seed değeri 1000-9999 aralığında olmalıdır

3.2. Metotlar

_initialize_state(): 3 farklı kaynağı birleştirerek başlangıç durumunu oluşturur. Dönüş tipi bytes.

generate_32bit(): 32-bit rastgele sayı üretir. Dönüş tipi int.

generate_hex_string(length): Hex formatında rastgele şifre üretir. Dönüş tipi str.

4. ÇALIŞMA PRENSİBİ

4.1. Entropi Kaynakları

Sistem üç farklı entropi kaynağını birleştirir:

Birinci kaynak - Kullanıcı Seed'i: 4 basamaklı kullanıcı girişi (1000-9999)
İkinci kaynak - Zaman Damgası: time.time_ns() ile nanosaniye hassasiyetinde zaman
Üçüncü kaynak - Sistem Rastgeleliği: os.urandom(16) ile 16 byte işletim sistemi rastgeleliği

4.2. Algoritma Akışı

Adım 1: Kullanıcı seed'i, zaman damgası ve os.urandom birleştirilir
Adım 2: Birleşik veri SHA-256 ile hashlenir
Adım 3: 32 byte başlangıç durumu elde edilir
Adım 4: Durum ve sayaç birleştirilerek tekrar SHA-256 uygulanır
Adım 5: Yeni durumun ilk 4 byte'ı 32-bit rastgele sayı olarak döndürülür

5.GÜVENLİK ÖZELLİKLERİ

5.1. Güçlü Yönleri

Çoklu Entropi Kaynakları: Tahmin edilebilirliği azaltır
SHA-256 Hash: Kriptografik olarak güvenli hash fonksiyonu
Sayaç Mekanizması: Aynı durumdan farklı çıktılar garantiler
Input Validasyonu: Seed değeri kontrol edilir

5.2. SHA-256 Kullanımının Avantajları

Tek yönlü fonksiyondur, tersine çevrilemez
Çakışma direnci yüksektir
256-bit çıktı uzunluğu sağlar
Her iterasyonda durum güncellenir

6. AVALANCHE ETKİSİ TESTİ

6.1. Testin Amacı
Birbirine çok yakın seed değerlerinin (örneğin 1234 ve 1235) tamamen farklı çıktılar ürettiğini doğrulamaktır.

6.2. Beklenen Sonuç
İdeal bit farklılık oranı yaklaşık yüzde 50'dir. Bu oran, hash fonksiyonunun kalitesinin göstergesidir. Yüksek oran iyi kriptografik özellik anlamına gelir.

7. KULLANIM ÖRNEKLERİ

7.1. Komut Satırından Çalıştırma
Terminalde python secure_random_generator.py komutu çalıştırılır.

7.2. Programatik Kullanım
Önce SecureRandomGenerator sınıfı import edilir. Ardından 4 basamaklı bir seed değeri ile üreteci başlatılır. generate_32bit() metodu ile 32-bit rastgele sayı üretilebilir. generate_hex_string(32) metodu ile 32 karakterlik hex formatında şifre üretilebilir.

ÖRNEK ÇIKTI

32-bit Rastgele Sayılar:
Birinci değer: 2847593621

İkinci değer: 1938274651
Üçüncü değer: 847291034

32 Karakterlik Hex Şifreler:
Birinci şifre: a9b3c5d573829abb32871c9a1e5f8d2c
İkinci şifre: 7f4e2d1c8b9a3e5f6c7d8e9f0a1b2c3d

SONUÇ:

Bu proje, modern kriptografik prensipleri kullanarak güvenli rastgele sayı üretimi sağlamaktadır. SHA-256 hash algoritması ve çoklu entropi kaynaklarının birleşimi, üretilen sayıların tahmin edilemez ve güvenli olmasını garantilemektedir.