

# Real Random

## Entropy-as-a-Service SDK

Mobile Security Integration Guide

Version 1.0 | October 2025

### Executive Overview

Real Random provides quantum-grade entropy generation using patented physical random number generation technology. This SDK enables mobile security applications to integrate True Random Number Generation (TRNG) for cryptographic operations that resist both classical and quantum computing attacks.

**Validated Performance:** Real Random has achieved an 88% entropy strength rating on NIST Statistical Test Suite testing with 0.999996 bits per byte entropy, independently validated by Rochester Institute of Technology. Performance is comparable to Random.org, the established commercial TRNG leader.

### Key Benefits for Mobile Security

- **Quantum-Secure Cryptography:** Physical entropy source eliminates vulnerabilities in algorithmic random number generators
- **Independently Validated:** 88% NIST entropy strength rating with academic validation from Rochester Institute of Technology
- **Commercial-Grade Performance:** 0.999996 bits per byte entropy - statistically comparable to Random.org
- **Zero Trust Architecture:** No reliance on device hardware RNGs that may be compromised
- **Observable Security:** Physical dice-based generation is transparent and verifiable, unlike black-box semiconductor noise
- **Low Latency:** Optimized for mobile networks with sub-100ms response times
- **Seamless Integration:** Drop-in replacement for standard random functions

### Architecture Overview

The Real Random Entropy-as-a-Service API operates as a RESTful service that delivers cryptographically secure random data generated from physical dice tumbling in viscous fluid. The entropy is captured via computer vision, digitized, and delivered through TLS-encrypted channels.

### How It Works

1. **Entropy Request:** Mobile app requests random bytes via HTTPS
2. **Physical Generation:** Physical dice are tumbled in viscous fluid, captured by high-speed cameras

**3. Entropy Extraction:** Computer vision extracts entropy from dice positions and rotations

**4. Quality Assurance:** Real-time statistical testing ensures entropy quality (NIST test methodologies)

**5. Secure Delivery:** Random bytes delivered via TLS 1.3 with perfect forward secrecy

## Quick Start Integration

### iOS (Swift)

#### Installation

```
pod 'RealRandomSDK', '~> 1.0'
```

#### Basic Usage

```
import RealRandomSDK

let client = RealRandomClient(apiKey: "YOUR_API_KEY")

// Generate 32 bytes of quantum-secure entropy
client.getEntropy(bytes: 32) { result in
    switch result {
    case .success(let data):
        // Use data for key generation
        let key = SymmetricKey(data: data)
    case .failure(let error):
        print("Error: \(error)")
    }
}
```

### Android (Kotlin)

#### Installation

```
dependencies {
    implementation 'com.realrandom:sdk:1.0.0'
}
```

#### Basic Usage

```
import com.realrandom.RealRandomClient

val client = RealRandomClient(apiKey = "YOUR_API_KEY")

// Generate 32 bytes of quantum-secure entropy
client.getEntropy(32) { result ->
    result.onSuccess { data ->
        // Use data for key generation
        val keySpec = SecretKeySpec(data, "AES")
    }
    result.onFailure { error ->
        Log.e("RealRandom", "Error: $error")
    }
}
```

## API Reference

### REST API Endpoint

**Base URL:** `https://api.realrandom.com/v1`

**Authentication:** API key in Authorization header

### Get Random Bytes

```
GET /entropy?bytes={count}
Authorization: Bearer YOUR_API_KEY
```

#### Parameters:

- **bytes:** Number of random bytes requested (1-1024)
- **format:** Optional. 'base64' or 'hex' (default: base64)

#### Response:

```
{
  "data": "base64_encoded_random_bytes",
  "bytes": 32,
  "entropy_bits": 256,
  "generated_at": "2025-10-29T12:34:56Z",
  "source": "physical_dice"
}
```

## Performance Specifications

Metric	Specification	Notes
Latency	< 100ms p95	US-based requests
Throughput	1000 requests/sec	Per API key
Entropy Quality	0.999996 bits/byte	Fourmilab ENT verified
NIST Testing	88% strength rating	NIST STS validated
Availability	99.9% SLA	Redundant generation systems
Security	TLS 1.3	Perfect forward secrecy

## Common Use Cases

### Secure Key Generation

Use Real Random entropy for generating cryptographic keys that will remain secure against quantum computing attacks. Ideal for long-term encrypted storage and secure communications.

## Session Token Creation

Generate unpredictable session tokens and authentication nonces that cannot be predicted by attackers with access to device hardware or system state.

## Password Salt Generation

Create unique salts for password hashing that are truly random and cannot be reproduced, even with knowledge of the device state or operating system.

## Secure Messaging

Generate per-message keys for end-to-end encrypted communications where perfect forward secrecy is critical for intelligence and national security applications.

## Implementation Best Practices

### Caching Strategy

For optimal performance, implement an entropy pool that pre-fetches random data during idle periods:

- Maintain a local pool of 1KB-4KB of unused entropy
- Refill pool when it drops below 25% capacity
- Use background fetch to maintain pool during app inactivity
- Never reuse entropy - consume and discard

### Error Handling

Implement graceful degradation for network failures:

- Retry failed requests with exponential backoff
- Fall back to device hardware RNG only if entropy pool is exhausted
- Log all fallback events for security audit
- Alert users when quantum-grade entropy is unavailable

### Security Considerations

- **API Key Storage:** Store API keys in platform keychain, never in code or preferences
- **Certificate Pinning:** Pin Real Random's TLS certificate to prevent MITM attacks
- **Memory Management:** Zero entropy buffers immediately after use
- **Audit Logging:** Log all entropy requests for security compliance

## Support & Resources

**Developer Portal:** <https://developers.realrandom.co>

**API Documentation:** <https://docs.realrandom.co/api>

**Test Results:** <https://realrandom.co/validation>

**Technical Support:** [support@realrandom.co](mailto:support@realrandom.co)

**Security Issues:** [security@realrandom.co](mailto:security@realrandom.co)

**GitHub Repository:** <https://github.com/realrandom/mobile-sdk>

## **Independent Testing & Validation**

- NIST Statistical Test Suite: 88% entropy strength rating
- Fourmilab ENT: 0.999996 bits per byte entropy
- DieHarder Suite: Comprehensive randomness validation
- Academic Validation: Rochester Institute of Technology
- Benchmark Testing: Performance comparable to Random.org

## **About Real Random**

Real Random LLC is a quantum-secure cryptography company based in Marco Island, Florida. Our patented physical random number generation technology serves intelligence community clients and has been recognized as a Gartner Cool Vendor in Data Security 2025.