
# iTech Labs Standards

## ITL-21

## Random Number Generator (RNG) standards

**Ver 1.2**

**Release date: March, 2021**

Table of Contents

# 1. Introduction

Random Number Generators (RNG) produce a sequence of numbers that cannot be reasonably predicted. The numbers are generated by Hardware Random Number Generator (HRNG), Pseudo Random Number Generator (HRNG), or a combination of both. While HRNGs generate numbers utilising constantly changing physical attributes of the hardware, PRNGs generate numbers utilising software algorithms.

While unpredictability is important, statistical qualities of the numbers are very important in a number of applications, especially gaming applications.

In most games/applications utilising random numbers, a scaling/shuffling method is used to produce smaller numbers from the very large numbers generated by the HRNG or PRNG. It is important that the scaling/shuffling process does not introduce any bias.

Certification of an HRNG is subject to the availability of extensive environmental test results from the manufacturer. Certification of a PRNG is subject to the use of a well-known public domain algorithm and the availability of the algorithm's code and its implementation.

# 2. General requirements

2.1 Raw random output from HRNG device or PRNG algorithm must:

- pass recognised statistical tests (e.g., Diehard, Dieharder, or NIST tests)
- be statistically independent
- be unpredictable, even for an intruder who is aware of the hardware logic or the software algorithm
- pass all tests during maximum load

# 3. Scaled/shuffled numbers

3.1 The scaling/shuffling process must not introduce any bias or any pattern of predictability.

3.2 The numbers must be uniformly distributed within both predetermined tolerance limits and the required range for the application.

3.3 The numbers must pass inferential statistical tests at 95% confidence level.

# 4. Hardware RNG's

4.1 There must be dynamic monitoring of output. In case of hardware RNG failure, there shall be an automatic failover to a standby RNG (which could be another HRNG or a backup software PRNG), or the game/application shall immediately shut down.

4.2 The HRNG has extensive environmental test results available from the manufacturer (published on their website or provided by them) and these results are satisfactory.

# 5. Software RNG's

5.1 The RNG algorithm must be a well-known and tested public domain algorithm.

5.2 The period of the RNG algorithm must be sufficiently large (e.g., $2^{64}$).

5.3 Seeding and re-seeding must be from a suitable entropy source. Any form of seeding or re-seeding shall not introduce predictability.

5.4 Re-seeding must not be a routine or regular practice. Re-seeding input is at least as random as the output of the RNG being re-seeded.

Continuous re-seeding is allowed if the algorithm is based on continuous re-seeding from multiple entropy sources.

5.5 The algorithm must be cryptographically secure where the game/application requires cryptographic security. A cryptographically secure algorithm may also be required to meet any regulatory requirements.

(In online gaming applications, statistical qualities are much more important than cryptographic security).

5.6 Background cycling of the RNG is required if the RNG is used by a single user or a limited number of users.

Background cycling of the RNG is not required where multiple game/application and users draw numbers from the same RNG (except where background cycling is a regulatory requirement).

5.7 Internal state of the RNG

- only the RNG module can access its internal state;
- if the internal state is saved to a persistent storage or transferred through a network, it is always encrypted;
- the internal state is thread-safe.

5.8 If the RNG is Operating System (OS) or library based:

- source code is available to verify the relevant requirements
- certification shall be restricted to the relevant OS version or library.

5.9 RNG code is in separate files/module, not part of the game/application code.

## 6. RNG number usage and storage

6.1 The scaled/shuffled numbers used by the game/application must not be affected by anything other than the values derived from the RNG.

6.2 When a game/application obtains a scaled/shuffled number, it must be used in accordance with the specifications and rules of the game/application and not discarded based on the value or outcome.

6.3 Mapping of scaled numbers must be linear (to preserve the linearity of randomness).

However, mapping to game symbols/application outcomes may not be linear where mapping follows a clearly defined logic (e.g., non-linear mapping such as use of a weighting table with unequal weights).

6.4 RNG output obtained by the game/application shall be applied immediately and in accordance with the applicable rules.

6.5 RNG output must not be stored external to the system memory (unless encrypted) before usage by the game/application.

Note: **Section 6** is applicable where certification of actual usage of random numbers is required. If section **Section 6** is excluded from testing, this shall be explicitly stated in the certification report.