rng labs

RNG Labs International Ltd.

46, Triq il-Parrocca
St. Venera
Malta

Tel + 356 79254072
www.rnglabs.net

Random Number Generator (RNG) Certification

Product name	Random Number Generator (RNG)
Jurisdiction	United Kingdom Gambling Commission (UKGC)
Applicant	MervCF Limited
Test institute	RNG Labs International Ltd.
Type of product	Random Number Generator (RNG)

Authorised by:

J. Micallef

John Micallef
Head of Lab
23rd December 2024

Table of contents

General data

Report number	ME-UK-001-2024
Jurisdiction	UK Gambling Commission
Regulations	UKGC remote gambling and software technical standards, published February 2021, updated May 2024.
Test date	14th October 2024
Project engineer	Mr. Loc Phan Van

Applicant data

Company name	MervCF Limited
Address	Av. Pastor Diaz, Provincia de Puntarenas, Jaco, 61101, Costa Rica
Contact	Dominika Pistorova

Platform information

Supplier	Value coders
Version	1.24.12.001

1. Introduction

The intent of this report is to indicate that RNG Labs International Ltd. has completed its evaluation of the Random Number Generator (RNG), version 1.24.12.001, provided by MervCF Limited.

2. Scope of testing

MervCF Limited submitted the required materials to RNG Labs in order to conduct a random number generator analysis on the RNG. The scope of this analysis was limited to software verification, source code review, and data analysis.

The RNG was evaluated against the RNG-specific requirements of the following technical standard:

- Testing strategy for compliance with remote gambling and software technical standards, April 2022.

3. Source code review

MervCF Limited submitted appropriate documentation and FULL source code which pertains to the generation of random numbers on 14th October 2024. RNG Labs reviewed the source code provided by tracing the path of the RNG application from the initiation of the draw to the selected output of random numbers.

RNG Labs inspected the source code, where practicable, in an attempt to find any undisclosed switches or parameters having a possible influence on randomness and fair play. RNG Labs assessed the ability of the RNG to produce all numbers within the desired range.

4. Test results overview

Requirements within this scope are included in this test results overview.

4.1 RNG details

RNG Description	The Random Number Generator (RNG) combines blockchain-based randomness from an Ethereum smart contract with local cryptographic XOR masking for enhanced security. Random numbers are retrieved from the contract, masked with secure 32-bit integers.
RNG Version No.	1.24.12.001
Hardware/Software Base	Software

Reference	Functionality	SHA-CheckSum
dieHardTest.js	RNG	23f3f7e56a4bb9e2516826b6d70611cec23ce3d8

4.2 Scope and approach to testing and a description of all tests applied

The scope of the RNG test is narrowly focused on a rigorous evaluation of the documentation, source code, and output of the RNG specifically against RTS Implementation Guidance 7A.a, ensuring compliance with recognised international standards. This testing aims to verify that:

1. The output of the RNG is uniformly distributed across the entire output range.
2. The outcomes of games using the RNG align with expected or theoretical probabilities.

This targeted approach emphasizes security, unpredictability, and non-repeatability in number generation while ensuring the RNG meets the criteria for acceptable randomness as defined by the applicable technical standards. Here’s a comprehensive detail of the scope and the approach to testing:

4.2.1 Documentation and code review

The license holder is required to submit documented references to the RNG’s algorithm, which should be well-established and published in a recognised international publication. We also require access to the source code and any related recalculative procedures linked to the RNG for an in-depth review. Our approach for this part of the scope will be as follows:

- Identifying RNG Algorithm: We will thoroughly identify the RNG algorithm and research any known weaknesses associated with it to assure its integrity and security.
- Verification of RNG Internal State: The internal state of the RNG will be scrutinised to confirm its adherence to unpredictability and non-repeatability requirements.
- Verification of RNG Implementation: We will verify seeding, background cycling, minimal reseeding to ensure that the RNG implementation caters to unpredictability and non-repeatability requirements efficiently.
- Verification of RNG Output Usage: The use of the random numbers, including scaling and shuffling, will be reviewed to ensure it aligns with industry standards.
- Compilation of RNG Code: After resolving any identified issues within the code, we will compile the RNG code to ensure its operability.

4.2.2 Testing RNG output:

- Diehard Test Suite: The RNG output will undergo the stringent Diehard test suite, which applies a series of statistical tests to determine the randomness of the output. These are a battery of statistical tests for measuring the quality of a RNG’s sequence of numbers and determining whether they are truly random.

- NIST Tests: The National Institute of Standards and Technology (NIST) tests will be applied to ensure the RNG output meets the national standard for randomness. These are designed to test the randomness of binary sequences produced by either hardware or software-based RNGs and consist of 15 different tests focused on various aspects of randomness.

Appendix A: Documentations & Code Review

RNG implementation

The code generates random numbers using a Solidity-based smart contract on the Ethereum blockchain. The RNG retrieves random outputs via the randomNumbersGenerate function of the smart contract, which is parameterized by a lottery ID and an initial array. To strengthen the randomness and mitigate potential predictability from the blockchain process, each random number is XOR-masked with a secure, locally generated 32-bit integer derived from the crypto module in Node.js. This combination ensures that the final output is resistant to manipulation and remains cryptographically secure.

- Blockchain-based Randomness: Utilizes the unpredictability of blockchain processes for generating random numbers.
- Cryptographic Enhancement: Applies an XOR mask with a secure random integer generated locally to reinforce randomness.
- Scalable Batch Processing: Efficiently processes and writes random numbers in batches to a binary file for further use or analysis.

Statistical analysis

In order to verify that the pseudo random numbers generated by the algorithm satisfy the 'acceptably random' requirement, the output of the RNG is subjected to a statistical analysis. This analysis consists of a series of tests that determine the chance that these numbers have not been generated by a random-like process. Each of these tests observes the behaviour of a specific aspect of the series of random numbers, and will fail if the chance that a random process has not generated these series is above a certain threshold.

These tests will verify whether the output from the RNG is uniformly distributed among the entire output range as stipulated by guidance rule 7A a. i), but is not limited to just this verification.

The software used for statistical analysis of raw output is the Dieharder RNG test suite (Brown, 2015). This is a test suite maintained by Robert G. Brown from Duke University Physics Department. It builds upon the Diehard battery of tests from George Marsaglia (Marsaglia, 1995), but also includes tests from the statistical test suite from NIST (Soto, 1999) and tests developed by Robert G. Brown himself.

The following "bitwise" tests from the NIST Test Suite were applied:

- Frequency (Monobits) Test
- Frequency Test within a Block
- Run Test
- Test for the Longest Run of Ones in a Block
- Binary Matrix Rank Test
- Discrete Fourier Transform (Spectral) Test
- Non-overlapping Template Matching Test
- Maurer's "Universal Statistical" Test
- Linear Complexity Test

- Serial Test
- Approximate Entropy Test
- Cumulative Sums (Cumsum) Test
- Random Excursions Test
- Random Excursions Variant Test

The following "bitwise" tests from the DIEHARD Battery of Tests of Randomness were applied:

- Birthday Spacing Test
- Overlapping 5-permutations Test
- Binary Rank 31×31 Test
- Binary Rank 32×32 Test
- Binary Rank 6×8 Test
- Bitstreams Test
- Overlapping Pairs Sparse Occupancy (OPSO) Test
- Overlapping Ouadruples Sparse Occupancy (OOSO) Test
- DNA Test
- Count the 1's (Specific Bytes) Test
- Count the 1's (Stream of Bytes) Test
- Parking Lot Test
- Minimum Distance Test
- 3-D Spheres Test
- Squeeze Test
- Overlapping Sums Test
- Runs Test
- Craps Test

The results of all the tests are listed in appendix B.

Source code inspection

The source code was inspected to verify that the remaining requirements 7A.a have been met. In this section for each of the requirements a brief outline is given how the source code ensures that the requirements are met.

The RNG is unpredictable

The RNG combines blockchain-based randomness with cryptographic XOR masking, ensuring unpredictability and resistance to manipulation.

The seeding is unpredictable

The unpredictability of the seeding in this RNG mechanism derives from two key factors:

- Blockchain-Based Randomness: The code retrieves random numbers from a smart contract deployed on the Ethereum blockchain. Blockchain-generated randomness often incorporates unpredictable factors such as block hashes, timestamps, and other on-chain data.
- Cryptographic XOR Masking: Each random number retrieved from the blockchain is further processed by applying a cryptographic XOR mask. This mask is generated

using the crypto module in Node.js, which relies on a cryptographically secure pseudorandom number generator (CSPRNG). The CSPRNG ensures that the XOR mask is highly random and unpredictable.

By combining these two independent sources of randomness, the code ensures that even if one source (e.g., the blockchain) is compromised or predictable to some extent, the overall RNG output remains secure and unpredictable.

The RNG does not cycle or synchronise

the RNG does not cycle or synchronize because it is based on independent and dynamic randomness sources.

Shuffling

The RNG performs shuffling-like behavior indirectly by introducing randomness into the array of numbers (initialTab) through the XOR masking process.

Scaling is applied properly

The RNG output is masked using XOR with a 32-bit cryptographically secure random integer, ensuring the results remain within the range of 32-bit unsigned integers (0 to 2^{32}-1). This maintains uniform distribution and prevents overflow or skewing in the generated random numbers.

Limitations

- Acceptable DoF: RNG provides outputs with high degrees of freedom, maintaining strong variability through blockchain randomness and cryptographic masking, suitable for cryptographic and non-deterministic applications.
- Usage: Suitable for generating independent random values; not designed for reuse of the same seed or synchronized environments.
- Security: Secure for cryptographic applications due to the combined use of blockchain randomness and cryptographically secure XOR masking. Scaling ensures uniform distribution without introducing bias.
- OS/System version and constraints: None.

This list was identified at the best of RNG Labs knowledge by analysing the test item(s) and collecting all possible reputable information sources at the time of the testing activity.

Appendix B: Empirical Testing

Please refer to the Appendices for details of the tests applied. The test results are summarised as follows:

NIST Test suite

The NIST Tests are based on the suite of tests released by the National Institute of Standards and Technology in Special Publication 800-22, Revision 1a (revised April 2010). They test sequences of raw binary output from the RNG. There were 03 sample size data file outputs tested, comprises of: 50.000, 100.000 and 150.000 random numbers.

Test name	P-Value (50.000)	P-Value (100.000)	P-Value (150.000)	Assessment
Frequency (Monobits) Test	0.474986	0.911413	0.213309	Pass
Frequency Test within a Block	0.534146	0.897763	0.637119	Pass
Run Test	0.554420	0.289667	0.494392	Pass
Test for the Longest Run of Ones in a Block	0.574903	0.983453	0.334538	Pass
Binary Matrix Rank Test	0.202268	0.171867	0.534146	Pass
Discrete Fourier Transform (Spectral) Test	0.494392	0.867692	0.262249	Pass
Non-overlapping Template Matching Test	0.616305	0.153763	0.699313	Pass
Maurer's "Universal Statistical" Test	0.595549	0.319084	0.657933	Pass
Linear Complexity Test	0.162606	0.924076	0.759756	Pass
Serial Test	0.699313	0.213309	0.455937	Pass
Approximate Entropy Test	0.474986	0.955835	0.798139	Pass
Cumulative Sums (Cumsum) Test	0.911413	0.946308	0.719747	Pass
Random Excursions Test	0.289667	0.888137	0.585209	Pass

Random Excursions Variant Test	0.289667	0.964295	0.723129	Pass

Conclusion: The RNG is **ACCEPTED** as random at the 95% confidence interval.

DIEHARD Battery of Tests

The Diehard Tests are based on the test suite published by George Marsaglia in 1995. They test sequences of raw binary output from the RNG. There were 03 sample size data file outputs tested, comprises of: 50.000, 100.000 and 150.000 random numbers.

Test name	P-values (50.000)	P-values (100.000)	P-values (150.000)	Assessment
Birthday Spacing Test	0.18433436	0.85352466	0.82778071	Pass
Overlapping 5-permutations Test	0.60637146	0.51022055	0.67641356	Pass
Binary Rank 32x32 Test	0.64182758	0.79802065	0.69771843	Pass
Binary Rank 6×8 Test	0.43675896	0.48657084	0.51818947	Pass
Bitstreams Test	0.49352527	0.78785676	0.78498908	Pass
Overlapping Pairs Sparse Occupancy (OPSO) Test	0.64666617	0.62063461	0.41302964	Pass
Overlapping Ouadruples Sparse Occupancy (OQSO) Test	0.94913788	0.59080907	0.57614654	Pass
DNA Test	0.09840402	0.42341150	0.45293128	Pass
Count the 1's (Specific Bytes) Test	0.28371487	0.98938570	0.74118503	Pass
Count the 1's (Stream of Bytes) Test	0.98396918	0.28344091	0.96404859	Pass
Parking Lot Test	0.71320308	0.87504329	0.53133973	Pass
Minimum Distance Test	0.66520395	0.16818337	0.17632933	Pass

3-D Spheres Test	0.43526243	0.36919213	0.31679052	Pass
Squeeze Test	0.14650376	0.65042615	0.88520634	Pass
Overlapping Sums Test	0.25586141	0.10228652	0.13961276	Pass
Runs Test	0.85446771	0.09187375	0.58181531	Pass
Craps Test	0.95938307	0.64400857	0.19638987	Pass

Conclusion: The RNG is **ACCEPTED** as random at the 95% confidence interval.