

Test report

Product name	: Random Number Generator (RNG)
Jurisdiction	: United Kingdom
Applicant	: Red Tiger Gaming Ltd.
Test institute	: Trisigma B.V.
Type of product	: Random Number Generator (RNG)

Author: M. Amigoni

Authorised by: D. Kuijer
Quality Manager

19-02-2019



Copyright © Trisigma B.V., Geldermalsen, the Netherlands. All rights reserved. The contents of this report may only be transmitted to third parties in its entirety. Application of the copyright notice and disclaimer is compulsory.

Trisigma B.V. disclaim liability for any direct, indirect, consequential or incidental damages that may result from the use of the information or data, or from the inability to use the information or data.

TABLE OF CONTENTS

1. TEST INSTITUTE	3
2. TEST METHODS.....	3
3. GENERAL REPORT DATA.....	4
4. APPLICANT DATA.....	4
5. CONCLUSION AND RECOMMENDATION	5
6. PLATFORM INFORMATION.....	5
7. REQUIREMENTS – TEST RESULTS OVERVIEW	6
APPENDIX A: RNG details and scope and approach to testing	10
APPENDIX B: Result of testing	13
APPENDIX C: Digital signature software version 1.0	16

1. TEST INSTITUTE

Trisigma B.V. (here after Trisigma) provides compliance and type approval services to the gaming industry and authorities. The Trisigma test labs are located in The Netherlands and have extensive facilities for testing and approval of online and land based gaming systems. Trisigma has been accredited by the Dutch Council of Accreditation for both standards ISO/IEC 17020 (with identification I254) and ISO/IEC 17025 (with identification L531) within the scope of compliance testing and examination of gaming systems. It is Trisigma's policy to carry out all activities according to these high quality standards in order to assure the international recognition of Trisigma certifications, reports and declarations.

This report presents the Trisigma final conclusion of compliance, the scope of examination, the specific identification of the gaming system and an overview of the applicable requirements including the appraisal with regard to the gaming system under examination.

This report has been constructed under the supervision and responsibility of Trisigma's Quality Manager. Every effort has been made to ensure the quality and accuracy of the information contained in this report. If errors or omissions are discovered, please contact us with details. Trisigma B.V. reserves the right to issue revisions of this test report if additional information is presented or discovered.

2. TEST METHODS

Trisigma examines gaming systems using accredited and recognized assessment methods. These methods cover all applicable components and characteristics of the product under examination.

Qualified test engineers carry out a comprehensive compilation of test methods using documentation review, measurements, evaluation of calculations and simulations, statistical tests, functional tests, visual assessment and source code analyses and supervised builds in order to examine the product from a requirements point of view. These test methods comprise the functional and statistical behavior of the gaming system.

3. GENERAL REPORT DATA

Report number	3s.18.514_UK-GI.R0
Jurisdiction	United Kingdom
Requirements	Remote gambling and software technical standards June 2017
Additional regulations or directions	Testing strategy for compliance with remote gambling and software technical standards, June 2017.
Test period	February 2019
Project Engineer	M. Amigoni
Revision information	Not applicable
References	Not applicable

4. APPLICANT DATA

Company name	Red Tiger Gaming Ltd.
Address	2nd Floor Athol House 21a - 23 Athol Street, Douglas IM1 1LB Isle of Man
Contact	Mrs Lisa Karran

5. CONCLUSION AND RECOMMENDATION

The RNG complies with the United Kingdom Remote gambling and software technical standards.

It is the recommendation of Trisigma that the RNG be approved for use in the jurisdiction of the United Kingdom.

The RNG has been tested according with the procedure for testing of the Testing strategy for compliance with remote gambling and software technical standards, June 2017.

NOTE

Software that meets the requirements of the UK RTOS is considered by the Gibraltar Gambling Commissioner's office to be compliant with the requirements of the Gibraltar RTOS.

6. PLATFORM INFORMATION

Supplier	Red Tiger Gaming Ltd.
Version	1.0.0

7. REQUIREMENTS – TEST RESULTS OVERVIEW

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

Test results overview		
Article	Requirement Text	Verdict
RTS requirement 7A	Random number generation and game results must be 'acceptably random'. Acceptably random here means that it is possible to demonstrate to a high degree of confidence that the output of the RNG, game, lottery and virtual event outcomes are random, through, for example, statistical analysis using generally accepted tests and methods of analysis. Adaptive behaviour (i.e. a compensated game) is not permitted.	PASS
	Remarks/Findings: Research demonstrates that events of chance are statistically random.	
RTS requirement 7A (continued)	Where lotteries use the outcome of other events external to the lottery, to determine the result of the lottery the outcome must be unpredictable and externally verifiable.	Not Applicable
	Remarks/Findings: This is not a lottery game.	
RTS implementation guidance 7A a.	RNG's should be capable of demonstrating the following qualities:	
	Remarks/Findings: This is an explanatory text only.	
RTS implementation guidance 7A a. i.	the output from the RNG is uniformly distributed over the entire output range and game, lottery, or virtual event outcomes are distributed in accordance with the expected/theoretical probabilities	
	Remarks/Findings: This is an explanatory text only.	
RTS implementation guidance 7A a. ii.	the output of the RNG, game, lottery, and virtual event outcomes should be unpredictable, for example, for a software RNG it should be computationally infeasible to predict what the next number will be without complete knowledge of the algorithm and seed value	
	Remarks/Findings: This is an explanatory text only.	
RTS implementation guidance 7A a. iii.	random number generation does not reproduce the same output stream (cycle), and that two instances of a RNG do not produce the same stream as each other (synchronise)	
	Remarks/Findings: This is an explanatory text only.	
RTS implementation guidance 7A a. iv.	any forms of seeding and re-seeding used do not introduce predictability	
	Remarks/Findings: This is an explanatory text only.	

RTS implementation guidance 7A a. v.	any scaling applied to the output of the random number generator maintains the qualities above.	
	Remarks/Findings: This is an explanatory text only.	
RTS implementation guidance 7A b.	For lotteries using external events - where it is not practical to demonstrate 7a. - the events outcomes should be:	
	Remarks/Findings: This is an explanatory text only.	
RTS implementation guidance 7A b. i.	unpredictable, that is, events should be selected only where they may reasonably be assumed to be random events	
	Remarks/Findings: This is an explanatory text only.	
RTS implementation guidance 7A b. ii.	unable to be influenced by the lottery operator (or external lottery manager)	
	Remarks/Findings: This is an explanatory text only.	
RTS implementation guidance 7A b. iii.	publicly available and externally verifiable, for example, events that are published in local or national press would be acceptable.	
	Remarks/Findings: This is an explanatory text only.	
RTS implementation guidance 7A c.	For games or virtual events that use the laws of physics to generate the outcome of the game (mechanical RNGs), the mechanical RNG used should be capable of meeting the requirements in a. where applicable and in addition:	
	Remarks/Findings: This is an explanatory text only.	
RTS implementation guidance 7A c. i.	the mechanical pieces should be constructed of materials to prevent decomposition of any component over time (e.g. a ball shall not disintegrate)	
	Remarks/Findings: This is an explanatory text only.	
RTS implementation guidance 7A c. ii.	the properties of physical items used to choose the selection should not be altered	
	Remarks/Findings: This is an explanatory text only.	
RTS implementation guidance 7A c. iii.	players should not have the ability to interact with, come into physical contact with, or manipulate the mechanics of the game.	
	Remarks/Findings: This is an explanatory text only.	
RTS implementation guidance 7A d.	Restricting adaptive behaviour prohibits automatic or manual interventions that change the probabilities of game outcomes occurring during play. Restricting adaptive behaviour is not intended to prevent games from offering bonus or special features that implement a different set of rules, if they are based on the occurrence of random events.	
	Remarks/Findings: This is an explanatory text only.	
RTS requirement 7B	As far as is reasonably possible, games and events must be implemented fairly and in accordance with the rules and prevailing payouts, where applicable, as they are described to the customer.	Not Applicable
	Remarks/Findings: Game implementation is outside the scope of this test report.	
RTS implementation guidance 7B a.	Games should implement the rules as described in the rules available to the customer before play commenced.	
	Remarks/Findings: This is an explanatory text only.	

RTS implementation guidance 7B b.	The mapping of the random inputs to game outcomes should be in accordance with prevailing probabilities, pay tables, etc.	
	Remarks/Findings: This is an explanatory text only.	
RTS implementation guidance 7B c.	When random numbers, scaled or otherwise, are received, e.g. following a game requesting a sequence of random numbers, they are to be used in the order in which they are received. For example, they may not be discarded due to adaptive behaviour.	
	Remarks/Findings: This is an explanatory text only.	
RTS implementation guidance 7B d.	Numbers or sequences of numbers are not to be discarded, unless they fall outside the expected range of numbers required by the virtual event – such an occurrence should result in an error being logged and investigated.	
	Remarks/Findings: This is an explanatory text only.	
RTS requirement 7C	Game designs or features that may reasonably be expected to mislead the customer about the likelihood of particular results occurring are not permitted, including substituting losing events with near-miss losing events and simulations of real devices that do not simulate the real probabilities of the device.	Not Applicable
	Remarks/Findings: Game implementation is outside the scope of this test report.	
RTS implementation guidance 7C a.	Where a virtual event simulates a physical device, the theoretical game probabilities should match the probabilities of the real device (for example, the probability of a coin landing heads must be 0.5 every time the coin is tossed).	
	Remarks/Findings: This is an explanatory text only.	
RTS implementation guidance 7C b.	Where multiple physical devices are simulated the probabilities of each outcome should be independent of the other simulated devices.	
	Remarks/Findings: This is an explanatory text only.	
RTS implementation guidance 7C c.	Games may not falsely display near-miss results, that is, the event may not substitute one losing outcome with a different losing outcome.	
	Remarks/Findings: This is an explanatory text only.	
RTS implementation guidance 7C d.	Where the event requires a pre-determined layout (for example, hidden prizes on a map), the locations of the winning spots should not change during play, except as provided for in the rules of the game.	
	Remarks/Findings: This is an explanatory text only.	
RTS implementation guidance 7C e.	Where games involve an element of skill, every outcome described in the virtual event rules or artwork should be possible, that is, the customer should have some chance of achieving an advertised outcome regardless of skill.	
	Remarks/Findings: This is an explanatory text only.	
RTS implementation guidance 7C f.	Where a customer contributes to a jackpot pool, that customer should be eligible to win the jackpot whilst they are playing that game, in accordance with the game and jackpot rules.	
	Remarks/Findings: This is an explanatory text only.	
RTS requirement 7D	The rules, payouts and outcome probabilities of a virtual event or game may not be changed while it is available for gambling, except as provided for in the rules of the game, lottery or virtual event. Such changes must be brought to customer's attention.	Not applicable
	Remarks/Findings: Game implementation is outside the scope of this test report.	

RTS implementation guidance 7D a.	Changes to game or event rules, paytables or other parameters that change the way in which a game, lottery, or event works, the winnings paid, or likelihood of winning (except as described in 7Dc), should be conducted with the game or event taken offline or suspended.	
	Remarks/Findings: This is an explanatory text only.	
RTS implementation guidance 7D b.	Altered games, lotteries, and events should display a notice that informs customers that the game or event has been changed, for example, 'rules changed', 'new odds', or 'different payouts'. The notice should be displayed on game selection screens and on the events themselves if it is possible for the customer to go straight to the event without using a selection screen.	
	Remarks/Findings: This is an explanatory text only.	
RTS implementation guidance 7D c.	This requirement is not intended to prevent games and virtual events where specified changes occur legitimately, in accordance with the game or event rules, for example:	
	Remarks/Findings: This is an explanatory text only.	
RTS implementation guidance 7D c. i.	virtual events, such as virtual racing products where the odds differ from event to event depending on the virtual runners	
	Remarks/Findings: This is an explanatory text only.	
RTS implementation guidance 7D c. ii.	virtual games, such as bingo where the odds of winning are dependent on the number of entrants	
	Remarks/Findings: This is an explanatory text only.	
RTS implementation guidance 7D c. iii.	games with progressive jackpots, where the amount that can be won changes over time	
	Remarks/Findings: This is an explanatory text only.	
RTS implementation guidance 7D c. iv.	games with bonus rounds where different rules apply, so long as these rounds are properly described to the customer	
	Remarks/Findings: This is an explanatory text only.	
RTS implementation guidance 7D c. v.	unspecified changes to rules, paytables or other parameters that change the way in which a game, lottery or event works are not permitted, for example, rules that state 'game rules may be changed at any time' would not be acceptable.	
	Remarks/Findings: This is an explanatory text only.	
RTS requirement 7E	Except in the case of subscription lotteries, the system clearly and accurately display the result of the game or event and the customer's gamble.	Not Applicable
	Remarks/Findings: Game presentation is outside the scope of this test report.	
RTS requirement 7E (continued)	The result must be displayed for a length of time that may reasonably be expected to be sufficient for the customer to understand the result of the game or event in the context of their gamble.	Not Applicable
	Remarks/Findings: Game presentation is outside the scope of this test report.	
RTS implementation guidance 7E	The game artwork and text should be sufficient to provide the customer with all of the information required to determine whether they have lost or won, and the value of any winnings.	
	Remarks/Findings: This is an explanatory text only.	

APPENDIX A: RNG details and scope and approach to testing

RTS requirement 7A

Random number generation and game results must be 'acceptably random'. Acceptably random here means that it is possible to demonstrate to a high degree of confidence that the output of the RNG, game, lottery and virtual event outcomes are random through, for example, statistical analysis using generally accepted tests and methods of analysis. Adaptive behaviour (i.e. a compensated game) is not permitted.

The standards document also provides several guidelines as to the methods that are to be used to evaluate the random number generators:

RTS implementation guidance 7A

- a. RNG's should be capable of demonstrating the following qualities:
 - i. the output from the RNG is uniformly distributed over the entire output range and game, lottery, or virtual event outcomes are distributed in accordance with the expected/theoretical probabilities
 - ii. the output of the RNG, game, lottery, and virtual event outcomes should be unpredictable, for example, for a software RNG it should be computationally infeasible to predict what the next number will be without complete knowledge of the algorithm and seed value
 - iii. random number generation does not reproduce the same output stream (cycle), and that two instances of a RNG do not produce the same stream as each other (synchronise)
 - iv. any forms of seeding and re-seeding used do not introduce predictability
 - v. any scaling applied to the output of the random number generator maintains the qualities above.

In the remainder of this document, these requirements are demonstrated by a statistical analysis of the output of the random number generator and an inspection of the source code.

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.

For statistical analysis of the output of the random number generator a file was created containing 1.25 billion symbols (for a total of 10GB) generated by the RNG. For this generation a test setup was used with an identical configuration as used in the production environment.

The main 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.

Results of the statistical analysis

The complete Dieharder suite was run using a sample of 1.25 billion random 64-bit integers (10 GB) generated with a platform that was similar to the production platform. The test suite consists of independent tests. For each test random numbers from the sample were used. Because the entire suite needs more than 10 billion random numbers, reuse of the random numbers from the data file was allowed, but not in the same test.

The RNG passed all statistical analysis applied to scaled symbols: uniform distribution and statistical independence applied on 3 sets of 10 million samples each on several representative ranges for many kinds of RNG driven games (0-36, 0-51, 0-99): all sample sets provided results in-line with chi-square tests.

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

All tests passed the first assessment.

Source code inspection

The source code was inspected to verify that the remaining requirements ii) - v) 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 interface to the games is through a wrapper. The analysis included source codes of the PHP framework used.

The RNG is unpredictable, does not cycle or synchronize

The RNG calls the native PHP (ver.7.1.9) `random_int()` function, which in turns relies on `getrandom(2)` syscall on Linux systems, as those used in the live environment of the Applicant (Debian 9.4/9.5, Kernel 4.9.0-6-amd64/4.9.0-8-amd64). By default, `getrandom()` draws entropy from the `urandom` source (i.e., the same source as the `/dev/urandom` device).

The underlying system algorithm is described at <https://eprint.iacr.org/2012/251.pdf> . The PHP interface produces 64-bit word length numbers. Period is estimated in $2^{92} \cdot 32 - 1$ cycle for the internal pools, while output pool has a period of $2^{26} \cdot 32 - 1$.

The `random_int()` function provides protection against prediction and backtracking attacks on the RNG, as it doesn't support direct manipulation of manual reseeding (restart of the system allows re-seeding) - the function generates cryptographically secure pseudo-random integers. If an appropriate source of randomness cannot be found, an Exception will be thrown.

This will ensure that the RNG will not cycle and does not synchronize.

The seeding is unpredictable

No public interfaces are available for re-seeding – the operation is performed frequently and autonomously at system level. The PHP implementation is designed to seed RNG instances at initialisation with several sources of entropy from the underlying operating system. However, if a seed file is saved across reboots the output is cryptographically secure against attackers without local root access as soon as it is reloaded in the boot sequence, and adequate for network encryption session keys. The kernel random-number generator is designed to produce a small amount of high-quality seed material to seed a cryptographic pseudo-random number generator (CPRNG), with specific design to guarantee security.

The reseeding strategies will in practice comply with the implementation guidance 7A. a. iv.

Scaling is applied properly

Scaling is performed within the `random_int()` method implementation. The implementation provides scaling functions that will scale the entropy buffer to 64 bit random numbers within a specific range of integers by calling with appropriate arguments, i.e. `random_int(min,max)`. The algorithm used for the scaling will discard any random numbers that may skew the distribution on the specified range.

No other functions are present to alter the results of the random number generator algorithm.

Limitations

- **Acceptable DoF:** any range with boundaries included within 64-bit integer limits. Range cannot exceed these values independently by the build architecture of PHP.
- **Usage:** suitable for usage with and without replacement.
- **Security:** suitable for cryptographic secure purposes.
- **OS/System version and constraints:** Linux Kernel 4.9, PHP 7.1.9.

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

APPENDIX B: Result of testing

```

=====#
#           dieharder version 3.31.1 Copyright 2003 Robert G. Brown           #
#=====#
# rng_name | filename | rands/second |
# mt19937 | output.bin | 1.91e+08 |
#=====#
# test_name | ntup | tsamples | psamples | p-value | Assessment |
#=====#
diehard_birthdays | 0 | 100 | 100 | 0.31414005 | PASSED
diehard_operm5 | 0 | 100000 | 100 | 0.45528802 | PASSED
diehard_rank_32x32 | 0 | 40000 | 100 | 0.90423434 | PASSED
diehard_rank_6x8 | 0 | 100000 | 100 | 0.61097984 | PASSED
diehard_bitstream | 0 | 2097152 | 100 | 0.98250673 | PASSED
diehard_opso | 0 | 2097152 | 100 | 0.96056393 | PASSED
diehard_oqso | 0 | 2097152 | 100 | 0.87984506 | PASSED
diehard_dna | 0 | 2097152 | 100 | 0.99137179 | PASSED
diehard_count_1s_str | 0 | 256000 | 100 | 0.62517099 | PASSED
diehard_count_1s_byt | 0 | 256000 | 100 | 0.69764347 | PASSED
diehard_parking_lot | 0 | 12000 | 100 | 0.79360041 | PASSED
diehard_2dsphere | 2 | 8000 | 100 | 0.86599281 | PASSED
diehard_3dsphere | 3 | 4000 | 100 | 0.63812348 | PASSED
diehard_squeeze | 0 | 100000 | 100 | 0.07031156 | PASSED
diehard_sums | 0 | 100 | 100 | 0.01559132 | PASSED
diehard_runs | 0 | 100000 | 100 | 0.24292641 | PASSED
diehard_runs | 0 | 100000 | 100 | 0.52137146 | PASSED
diehard_craps | 0 | 200000 | 100 | 0.29296065 | PASSED
diehard_craps | 0 | 200000 | 100 | 0.79571458 | PASSED
marsaglia_tsang_gcd | 0 | 10000000 | 100 | 0.76374817 | PASSED
marsaglia_tsang_gcd | 0 | 10000000 | 100 | 0.89658615 | PASSED
sts_monobit | 1 | 100000 | 100 | 0.08674300 | PASSED
sts_runs | 2 | 100000 | 100 | 0.98131148 | PASSED
sts_serial | 1 | 100000 | 100 | 0.34744976 | PASSED
sts_serial | 2 | 100000 | 100 | 0.91015257 | PASSED
sts_serial | 3 | 100000 | 100 | 0.28641626 | PASSED
sts_serial | 3 | 100000 | 100 | 0.55960092 | PASSED
sts_serial | 4 | 100000 | 100 | 0.39004565 | PASSED
sts_serial | 4 | 100000 | 100 | 0.39219043 | PASSED
sts_serial | 5 | 100000 | 100 | 0.67181068 | PASSED
sts_serial | 5 | 100000 | 100 | 0.57561839 | PASSED
sts_serial | 6 | 100000 | 100 | 0.54721635 | PASSED
sts_serial | 6 | 100000 | 100 | 0.07069975 | PASSED
sts_serial | 7 | 100000 | 100 | 0.40890395 | PASSED
sts_serial | 7 | 100000 | 100 | 0.22859399 | PASSED
sts_serial | 8 | 100000 | 100 | 0.79550377 | PASSED

```

sts_serial	8	100000	100 0.74937484	PASSED
sts_serial	9	100000	100 0.87090103	PASSED
sts_serial	9	100000	100 0.02189947	PASSED
sts_serial	10	100000	100 0.56485792	PASSED
sts_serial	10	100000	100 0.14749371	PASSED
sts_serial	11	100000	100 0.27818879	PASSED
sts_serial	11	100000	100 0.83962850	PASSED
sts_serial	12	100000	100 0.77767626	PASSED
sts_serial	12	100000	100 0.27756248	PASSED
sts_serial	13	100000	100 0.71651946	PASSED
sts_serial	13	100000	100 0.37959161	PASSED
sts_serial	14	100000	100 0.97794612	PASSED
sts_serial	14	100000	100 0.80285497	PASSED
sts_serial	15	100000	100 0.92790407	PASSED
sts_serial	15	100000	100 0.50982585	PASSED
sts_serial	16	100000	100 0.31482640	PASSED
sts_serial	16	100000	100 0.43671275	PASSED
rgb_bitdist	1	100000	100 0.74396875	PASSED
rgb_bitdist	2	100000	100 0.04319113	PASSED
rgb_bitdist	3	100000	100 0.04455212	PASSED
rgb_bitdist	4	100000	100 0.11027802	PASSED
rgb_bitdist	5	100000	100 0.29378143	PASSED
rgb_bitdist	6	100000	100 0.13734854	PASSED
rgb_bitdist	7	100000	100 0.05830055	PASSED
rgb_bitdist	8	100000	100 0.96710406	PASSED
rgb_bitdist	9	100000	100 0.87401289	PASSED
rgb_bitdist	10	100000	100 0.78901064	PASSED
rgb_bitdist	11	100000	100 0.87288081	PASSED
rgb_bitdist	12	100000	100 0.35142804	PASSED
rgb_minimum_distance	2	10000	1000 0.50062475	PASSED
rgb_minimum_distance	3	10000	1000 0.40921092	PASSED
rgb_minimum_distance	4	10000	1000 0.77448765	PASSED
rgb_minimum_distance	5	10000	1000 0.75003451	PASSED
rgb_permutations	2	100000	100 0.94947256	PASSED
rgb_permutations	3	100000	100 0.72573414	PASSED
rgb_permutations	4	100000	100 0.13397159	PASSED
rgb_permutations	5	100000	100 0.65321333	PASSED
rgb_lagged_sum	0	1000000	100 0.86785065	PASSED
rgb_lagged_sum	1	1000000	100 0.09158888	PASSED
rgb_lagged_sum	2	1000000	100 0.96020526	PASSED
rgb_lagged_sum	3	1000000	100 0.66695755	PASSED
rgb_lagged_sum	4	1000000	100 0.94991947	PASSED
rgb_lagged_sum	5	1000000	100 0.20019230	PASSED
rgb_lagged_sum	6	1000000	100 0.64845659	PASSED
rgb_lagged_sum	7	1000000	100 0.11645969	PASSED

```

rgb_lagged_sum| 8| 1000000| 100|0.80425999| PASSED
rgb_lagged_sum| 9| 1000000| 100|0.07159222| PASSED
rgb_lagged_sum| 10| 1000000| 100|0.31790021| PASSED
rgb_lagged_sum| 11| 1000000| 100|0.57203830| PASSED
rgb_lagged_sum| 12| 1000000| 100|0.98587689| PASSED
rgb_lagged_sum| 13| 1000000| 100|0.17897525| PASSED
rgb_lagged_sum| 14| 1000000| 100|0.64915046| PASSED
rgb_lagged_sum| 15| 1000000| 100|0.19423938| PASSED
rgb_lagged_sum| 16| 1000000| 100|0.36611468| PASSED
rgb_lagged_sum| 17| 1000000| 100|0.01634313| PASSED
rgb_lagged_sum| 18| 1000000| 100|0.54438752| PASSED
rgb_lagged_sum| 19| 1000000| 100|0.63284327| PASSED
rgb_lagged_sum| 20| 1000000| 100|0.45256047| PASSED
rgb_lagged_sum| 21| 1000000| 100|0.02236337| PASSED
rgb_lagged_sum| 22| 1000000| 100|0.70865986| PASSED
rgb_lagged_sum| 23| 1000000| 100|0.41506609| PASSED
rgb_lagged_sum| 24| 1000000| 100|0.92103146| PASSED
rgb_lagged_sum| 25| 1000000| 100|0.15173183| PASSED
rgb_lagged_sum| 26| 1000000| 100|0.35985868| PASSED
rgb_lagged_sum| 27| 1000000| 100|0.38141382| PASSED
rgb_lagged_sum| 28| 1000000| 100|0.18838365| PASSED
rgb_lagged_sum| 29| 1000000| 100|0.65551712| PASSED
rgb_lagged_sum| 30| 1000000| 100|0.51234140| PASSED
rgb_lagged_sum| 31| 1000000| 100|0.93069776| PASSED
rgb_lagged_sum| 32| 1000000| 100|0.05484490| PASSED
rgb_kstest_test| 0| 10000| 1000|0.17097378| PASSED
dab_bytedistrib| 0| 51200000| 1|0.83667853| PASSED
dab_dct| 256| 50000| 1|0.98099287| PASSED
Preparing to run test 207. ntuple = 0
dab_filltree| 32| 15000000| 1|0.69506355| PASSED
dab_filltree| 32| 15000000| 1|0.19104132| PASSED
Preparing to run test 208. ntuple = 0
dab_filltree2| 0| 5000000| 1|0.31804558| PASSED
dab_filltree2| 1| 5000000| 1|0.68295132| PASSED
Preparing to run test 209. ntuple = 0
dab_monobit2| 12| 65000000| 1|0.93934769| PASSED

```

APPENDIX C: Digital signature software version 1.0

File name	SHA1
Crypto.php	bda2560dd71adfcc0324cc7e94f174bfc249ee68
ext/standard/basic_functions.c	9285cf2dc570cab67af5808ab157bfd77339809c
ext/standard/php_random.h	963596cc46ce317361885b1b313766b582bada32
ext/standard/random.c	a55df4c579e65527b73ddfbe08f3bff8b28d5ca5