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

Author: D. Zoer

Authorised by: Ing. R. Hubregtse 07-04-2017
Quality Manager

TABLE OF CONTENTS

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 comprises the functional and statistical behavior of the gaming system.

3. GENERAL REPORT DATA

Report number	3s.16.052_UK_GI.R1
Jurisdiction	United Kingdom
Requirements	Remote gambling and software technical standards July 2015
Additional regulations or directions	Testing strategy for compliance with remote gambling and software technical standards, first published August 2009, updated July 2015.
Test period	July - August 2015
Project Engineer	D. Zoer
Revision information	This R1 revision supersedes report 3s.16.052_UK.R0. The conclusion is expanded with a note on the Gibraltar Gambling Commissioner's office consideration on compliance. Company name changed from TGP Games Ltd. to Red Tiger Gaming Ltd. Holder of platform supplier name changed from TGP Games Ltd. to Red Tiger Gaming Ltd.
References	-

4. APPLICANT DATA

Company name	Red Tiger Gaming Ltd.
Address	2nd Floor Armitage House IM1 5AW, Ridgeway Street Douglas Isle of Man
Contact	Mrs Lisa Karran

5. CONCLUSION AND RECOMMENDATION

The Random Number Generator (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 level 1 definition of the Testing Strategy for compliance with remote gambling and software technical standards, first published August 2009, updated July 2015.

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

Platform information:

Supplier	Red Tiger Gaming Ltd.
Version	1.0

7. REQUIREMENTS – TEST RESULTS OVERVIEW

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

Test results overview		
Article	**Requirement Text**	**Verdict**
RTS aim 7	To ensure that games and other virtual events operate fairly.	
	Remarks/Findings: This is an explanatory text only.	
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.	**PASS**
	Remarks/Findings: The RNG and mapping and scaling is implemented fairly.	
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.	**PASS**
	Remarks/Findings: The RNG probabilities do not change.	
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	games with progressive jackpots, where the amount that can be won changes over time	

c. iii.	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 must be designed to 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 10 billion 32 bit random numbers generated by the RNG. For this generation a test setup was used with an identical configuration as used in the production environment.

The software used for statistical analysis 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 10 billion random numbers 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 results of all the tests are listed in appendix B.

All tests but two passed the first assessment. These two tests yielded inconclusive results, signified with the *WEAK* predicate in the tests results. Those tests were run again, with a command line option that makes the test run until an unambiguous result is obtained. All tests subsequently passed.

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 is unpredictable

The RNG calls the native PHP (ver.5.6) mt_rand() function. The algorithm is the Mersenne Twister and produces 32-bit word length numbers. The cycle length is a very long period of $2^{19937} – 1$. The test results show the RNG to be unpredictable random. The mt_rand() function will provide protection against prediction and backtracking attacks on the RNG, if timely reseeded with enough entropy.

The seeding is unpredictable

The PHP implementation is designed to seed itself at initialisation, the RNG will be seeded with a source of entropy from the underlying operating system. Subsequent reseeding uses LCG. The code allows for two reseeding strategies:

1. Reseeding at each start of the httpd process.
2. Reseeding for each new pid, for multithreaded httpd.

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

The RNG does not cycle or synchronize

The PHP mt-Rand() implementation has 2^{19937} – 1 internal states. The wrapper code will re-seed with each start of httpd and each new pid, as demonstrated in the previous section. This will ensure that the RNG will not cycle and does not synchronize.

Scaling is applied properly

Scaling is performed within the mt_rand() implementation. The implementation provides scaling functions that will scale 32 bit random numbers to a specified range of integers by calling with appropriate arguments, mt_rand(min,max). The algorithm used for the scaling will discard any 32 bit 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.

APPENDIX B: Result of testing

Complete test run

```
#=============================================================================#
#            dieharder version 3.31.1 Copyright 2003 Robert G. Brown          #
#=============================================================================#
   rng_name    |           filename             |rands/second|
 file_input_raw|    TGP-PHP-MT-10GB.bin         |  1.10e+07  |
#=============================================================================#
        test_name   |ntup| tsamples |psamples|  p-value |Assessment
#=============================================================================#
   diehard_birthdays|   0|       100|     100|0.65272686|  PASSED
      diehard_operm5|   0|   1000000|     100|0.24717054|  PASSED
  diehard_rank_32x32|   0|     40000|     100|0.14359801|  PASSED
    diehard_rank_6x8|   0|    100000|     100|0.86005740|  PASSED
   diehard_bitstream|   0|   2097152|     100|0.25546788|  PASSED
        diehard_opso|   0|   2097152|     100|0.16901688|  PASSED
        diehard_oqso|   0|   2097152|     100|0.08739302|  PASSED
         diehard_dna|   0|   2097152|     100|0.13700885|  PASSED
diehard_count_1s_str|   0|    256000|     100|0.96957619|  PASSED
diehard_count_1s_byt|   0|    256000|     100|0.13644141|  PASSED
 diehard_parking_lot|   0|     12000|     100|0.31640509|  PASSED
    diehard_2dsphere|   2|      8000|     100|0.41415859|  PASSED
    diehard_3dsphere|   3|      4000|     100|0.47686160|  PASSED
     diehard_squeeze|   0|    100000|     100|0.15154088|  PASSED
        diehard_sums|   0|       100|     100|0.01905141|  PASSED
        diehard_runs|   0|    100000|     100|0.43778427|  PASSED
        diehard_runs|   0|    100000|     100|0.31500938|  PASSED
       diehard_craps|   0|    200000|     100|0.88744041|  PASSED
       diehard_craps|   0|    200000|     100|0.32182744|  PASSED
 marsaglia_tsang_gcd|   0|  10000000|     100|0.04400467|  PASSED
 marsaglia_tsang_gcd|   0|  10000000|     100|0.03777902|  PASSED
         sts_monobit|   1|    100000|     100|0.20654255|  PASSED
            sts_runs|   2|    100000|     100|0.93628125|  PASSED
          sts_serial|   1|    100000|     100|0.90148564|  PASSED
          sts_serial|   2|    100000|     100|0.93686837|  PASSED
          sts_serial|   3|    100000|     100|0.15985003|  PASSED
          sts_serial|   3|    100000|     100|0.03432608|  PASSED
          sts_serial|   4|    100000|     100|0.30469496|  PASSED
          sts_serial|   4|    100000|     100|0.56429901|  PASSED
          sts_serial|   5|    100000|     100|0.02456977|  PASSED
          sts_serial|   5|    100000|     100|0.17380272|  PASSED
          sts_serial|   6|    100000|     100|0.10045982|  PASSED
          sts_serial|   6|    100000|     100|0.88642243|  PASSED
          sts_serial|   7|    100000|     100|0.59116228|  PASSED
          sts_serial|   7|    100000|     100|0.39875714|  PASSED
          sts_serial|   8|    100000|     100|0.57181410|  PASSED
          sts_serial|   8|    100000|     100|0.27513836|  PASSED
          sts_serial|   9|    100000|     100|0.20971762|  PASSED
          sts_serial|   9|    100000|     100|0.10182367|  PASSED
          sts_serial|  10|    100000|     100|0.14163431|  PASSED
          sts_serial|  10|    100000|     100|0.41049159|  PASSED
          sts_serial|  11|    100000|     100|0.33174117|  PASSED
          sts_serial|  11|    100000|     100|0.97506164|  PASSED
          sts_serial|  12|    100000|     100|0.81345698|  PASSED
          sts_serial|  12|    100000|     100|0.27991975|  PASSED
          sts_serial|  13|    100000|     100|0.24808765|  PASSED
```

```
          sts_serial|  13|    100000|     100|0.62937248|  PASSED
          sts_serial|  14|    100000|     100|0.48205623|  PASSED
          sts_serial|  14|    100000|     100|0.99305835|  PASSED
          sts_serial|  15|    100000|     100|0.96945496|  PASSED
          sts_serial|  15|    100000|     100|0.20927608|  PASSED
          sts_serial|  16|    100000|     100|0.91040862|  PASSED
          sts_serial|  16|    100000|     100|0.85728446|  PASSED
         rgb_bitdist|   1|    100000|     100|0.68373697|  PASSED
         rgb_bitdist|   2|    100000|     100|0.73353601|  PASSED
         rgb_bitdist|   3|    100000|     100|0.41442498|  PASSED
         rgb_bitdist|   4|    100000|     100|0.58748365|  PASSED
         rgb_bitdist|   5|    100000|     100|0.42611181|  PASSED
         rgb_bitdist|   6|    100000|     100|0.68612815|  PASSED
         rgb_bitdist|   7|    100000|     100|0.85202214|  PASSED
         rgb_bitdist|   8|    100000|     100|0.70986178|  PASSED
         rgb_bitdist|   9|    100000|     100|0.90056338|  PASSED
         rgb_bitdist|  10|    100000|     100|0.64662684|  PASSED
         rgb_bitdist|  11|    100000|     100|0.48256567|  PASSED
         rgb_bitdist|  12|    100000|     100|0.33810644|  PASSED
rgb_minimum_distance|   2|     10000|    1000|0.38070421|  PASSED
rgb_minimum_distance|   3|     10000|    1000|0.40193277|  PASSED
rgb_minimum_distance|   4|     10000|    1000|0.44576631|  PASSED
rgb_minimum_distance|   5|     10000|    1000|0.51688039|  PASSED
    rgb_permutations|   2|    100000|     100|0.99551130|   WEAK
    rgb_permutations|   3|    100000|     100|0.59617502|  PASSED
    rgb_permutations|   4|    100000|     100|0.33588292|  PASSED
    rgb_permutations|   5|    100000|     100|0.25417855|  PASSED
      rgb_lagged_sum|   0|   1000000|     100|0.98272210|  PASSED
      rgb_lagged_sum|   1|   1000000|     100|0.88513224|  PASSED
      rgb_lagged_sum|   2|   1000000|     100|0.17118153|  PASSED
      rgb_lagged_sum|   3|   1000000|     100|0.39417413|  PASSED
      rgb_lagged_sum|   4|   1000000|     100|0.98288537|  PASSED
      rgb_lagged_sum|   5|   1000000|     100|0.98810637|  PASSED
      rgb_lagged_sum|   6|   1000000|     100|0.56635412|  PASSED
      rgb_lagged_sum|   7|   1000000|     100|0.60478010|  PASSED
      rgb_lagged_sum|   8|   1000000|     100|0.16329481|  PASSED
      rgb_lagged_sum|   9|   1000000|     100|0.18484039|  PASSED
      rgb_lagged_sum|  10|   1000000|     100|0.26659464|  PASSED
      rgb_lagged_sum|  11|   1000000|     100|0.73239241|  PASSED
      rgb_lagged_sum|  12|   1000000|     100|0.46732258|  PASSED
      rgb_lagged_sum|  13|   1000000|     100|0.95971924|  PASSED
      rgb_lagged_sum|  14|   1000000|     100|0.68792786|  PASSED
      rgb_lagged_sum|  15|   1000000|     100|0.45966489|  PASSED
      rgb_lagged_sum|  16|   1000000|     100|0.05101945|  PASSED
      rgb_lagged_sum|  17|   1000000|     100|0.73404103|  PASSED
      rgb_lagged_sum|  18|   1000000|     100|0.89349244|  PASSED
      rgb_lagged_sum|  19|   1000000|     100|0.99706115|   WEAK
      rgb_lagged_sum|  20|   1000000|     100|0.92740161|  PASSED
      rgb_lagged_sum|  21|   1000000|     100|0.24152772|  PASSED
      rgb_lagged_sum|  22|   1000000|     100|0.08244362|  PASSED
      rgb_lagged_sum|  23|   1000000|     100|0.11789208|  PASSED
      rgb_lagged_sum|  24|   1000000|     100|0.49152105|  PASSED
      rgb_lagged_sum|  25|   1000000|     100|0.84530885|  PASSED
      rgb_lagged_sum|  26|   1000000|     100|0.30332553|  PASSED
      rgb_lagged_sum|  27|   1000000|     100|0.42756325|  PASSED
      rgb_lagged_sum|  28|   1000000|     100|0.97189555|  PASSED
      rgb_lagged_sum|  29|   1000000|     100|0.78977948|  PASSED
      rgb_lagged_sum|  30|   1000000|     100|0.93662820|  PASSED
```

```
      rgb_lagged_sum|  31|   1000000|     100|0.27159178|  PASSED
      rgb_lagged_sum|  32|   1000000|     100|0.84013380|  PASSED
     rgb_kstest_test|   0|     10000|    1000|0.44644557|  PASSED
     dab_bytedistrib|   0|  51200000|       1|0.89554481|  PASSED
             dab_dct| 256|     50000|       1|0.16940695|  PASSED
        dab_filltree|  32|  15000000|       1|0.81358839|  PASSED
        dab_filltree|  32|  15000000|       1|0.08975299|  PASSED
       dab_filltree2|   0|   5000000|       1|0.30496585|  PASSED
       dab_filltree2|   1|   5000000|       1|0.53650538|  PASSED
        dab_monobit2|  12|  65000000|       1|0.89087271|  PASSED
```

Rerun rgb_permutations

```
#=============================================================================#
#            dieharder version 3.31.1 Copyright 2003 Robert G. Brown          #
#=============================================================================#
   rng_name    |           filename             |rands/second|
 file_input_raw|    TGP-PHP-MT-10GB.bin         |  1.10e+07  |
#=============================================================================#
        test_name   |ntup| tsamples |psamples|  p-value |Assessment
#=============================================================================#
    rgb_permutations|   5|    100000|     100|0.69231327|  PASSED
```

Rerun rgb_lagged_sum

```
#=============================================================================#
#            dieharder version 3.31.1 Copyright 2003 Robert G. Brown          #
#=============================================================================#
   rng_name    |           filename             |rands/second|
 file_input_raw|    TGP-PHP-MT-10GB.bin         |  1.10e+07  |
#=============================================================================#
        test_name   |ntup| tsamples |psamples|  p-value |Assessment
#=============================================================================#
      rgb_lagged_sum|   0|   1000000|     100|0.52697621|  PASSED
```

APPENDIX C: Software digital signature

File name	SHA-256
PHP package 5.6.7-1	ac37b74c00ccefda94ce79a1212d827d1185b655a3c4ac7e88b937834dd5809b