Cryptographic Randomness Testing of Block Ciphers and Hash Functions

Ali Doğanaksoy, Barış Ege, Onur Koçak and Fatih Sulak

Institute of Applied Mathematics,
Middle East Technical University, Ankara, Turkey
{aldoks,baris.ege,onur.kocak,sulak}@metu.edu.tr

Abstract. One of the most basic properties expected from block ciphers and hash functions is passing statistical randomness testing, as they are expected to behave like random mappings. Previously, testing of AES candidate block ciphers was done by concatenating the outputs of the algorithms obtained from various input types. In this work, a more convenient method, namely the cryptographic randomness testing is introduced. A package of statistical tests are designed based on certain cryptographic properties of block ciphers and hash functions to evaluate their randomness. The package contains some new test functions beside the adaptation of some others which are introduced before. Also, the application of the package to the AES finalists produced more precise results than those obtained in similar applications.

1 Introduction

One of the basic properties of cryptographic primitives such as block ciphers and hash functions is indistinguishability from a random mapping. Therefore, the evaluation of the outputs of the algorithms by means of statistical randomness tests is of great importance. This process consists of two parts: taking a sample sequence from the algorithm, and analyzing this sample by statistical randomness tests.

During AES (Advanced Encryption Standard) competition, statistical testing of the candidate block ciphers was done by J. Soto, using the NIST Test Suite[1]. However, some of the tests in the suite require sequences of length at least 10^6 bits, where the block size of the AES candidates were 128 bits. Soto overcame this problem by performing statistical analysis to the concatenation of the outputs of the candidate algorithms, which correspond to various input types such as key avalanche, high weight, low weight, and the like[2].

In addition to the randomness of their outputs, block ciphers and hash functions should satisfy certain cryptographic properties as well. When block ciphers are considered, diffusion and confusion are among the most important properties, while for hash functions one of the basic design criteria is collision resistance. The cryptographic properties mentioned in this paper are as follows:

- Whenever one input bit is changed, every output bit should change with probability a half to achieve ideal diffusion. This criterion is called the strict avalanche criterion (SAC).
- The distance of a boolean function to the set of all affine functions should be large. This property is measured in terms of nonlinearity, and it is a concept related to confusion.
- Finding two inputs that have the same output should be hard, which is called the collision resistance property.
- Block ciphers with a fixed plaintext and hash functions are one way functions and they are required to behave like a random mapping.

Corresponding to these four cryptographic criteria, four randomness tests for block ciphers and hash functions are considered:

- The aim of the SAC Test is to measure if an algorithm satisfies the SAC property.
- The Linear Span Test evaluates an algorithm by examining the linear dependence of the outputs formed from a highly linearly dependent set of inputs.
- The subject of the Collision Test is the number of collisions in a portion of the output corresponding to a random subset of the input set.
- Coverage Test takes a subset of the input set and examines the size of the corresponding output set.

The structure of the work is as follows: in section 2, the details of cryptographic randomness tests used in the work are given. In section 3, the cryptographic randomness tests are applied to AES finalist algorithms and the results are compared to the ones given in [11] by Soto and Bassham. In the last section, conclusion and possible future work ideas are given.

2 Statistical and Structural Tests

Statistical randomness tests are functions that take arbitrary length input and produce a real number between 0 and 1 called the p-value, which is produced by evaluating certain randomness properties of the given input. For example, Frequency Test produces p-values depending on the number of ones in a binary sequence, where Overlapping Template Test evaluates the number of occurrences of a specific sequence of bits [1].

On the other hand, cryptographic randomness tests are a kind of statistical test that evaluate a function through investigating its certain cryptographic properties. In this section SAC Test, Linear Span Test, Collision Test and Coverage Test are defined in detail. In each subsection, the general idea, the mathematical background and application details of individual tests are given. For a better understanding of the concepts, it is assumed that the function under test is a function $f : F_2^n \times F_2^m \mapsto F_2^n$, where in the case of block cipher, n stands for the block size, and m stands for the key size. Similarly for the case of hash function, m and n stand for the message block size and the chaining variable size respectively.

2.1 SAC Test

The abbreviation SAC stands for strict avalanche criterion which was originally proposed for s-boxes by Webster and Tavares in 1986 [3]. SAC states that for a particular s-box, whenever one input bit is changed, every output bit must change with probability $\frac{1}{2}$. Therefore, extending this idea to the round functions of block ciphers (or the compression functions of hash functions) as previously done for stream ciphers [4], this test evaluates the given function by examining the effect of a single bit flip on the output bits. To achieve this, an $m \times n$ matrix called the *SAC Matrix* is formed in the following way: first all entries of the *SAC Matrix* are set to 0, a random input is taken and the output is computed. Then, after flipping the i^{th} bit of the input, the corresponding output is XORed to the original output. Afterwards, for each non-zero bit j of the output, $(i, j)^{\text{th}}$ entry of the *SAC Matrix* is incremented by 1. This is done for each input bit i, and the whole process is repeated for 2^{20} different random inputs[1]. Here, let K be the number of hits that an entry get, then

$$Pr(K = k) = \frac{\binom{n}{k}}{2^n} .$$

Therefore the expected value of each entry of the matrix is 2^{19}, and the distribution of the values of the matrix should follow a binomial distribution.

Following the construction of the *SAC Matrix*, χ^2 Goodness of Fit Test with the probabilities derived from Table 1 is used to evaluate the distribution of the values of the entire matrix. If a matrix produces a p-value less than 0.01, then it is considered non-random [4].

This method may fail to catch the correlation between a particular input and a particular output bit since the matrix is evaluated as a whole, therefore another method is proposed to evaluate each entry in the matrix. The entries outside a specific interval are flagged. The expected interval is taken as $[2^{19} - 5009, 2^{19} + 5009]$, and is computed so that a 2^{20}-bit sequence with a weight out of this

[1] This process should be repeated as many times as possible to detect small biases. If the performance of the function under testing is not good, the process is repeated for a less number of random inputs. Also, if the process is repeated more than 2^{20} times, the probabilities should be calculated with more than 6 digit precision.

Table 1. Ranges and probabilities of SAC Test for 2^{20} trials

Bin	Range	Probability
1	0-523857	0.200224
2	523858-524158	0.199937
3	524159-524417	0.199677
4	524418-524718	0.199937
5	524719-1048576	0.200224

interval would be assigned a p-value less than 10^{-6} from the Frequency Test. Since in the *SAC Matrix*, the number of entries is close to 10^6, it is not much unexpected to observe a term smaller than 10^{-6}. Therefore, the test is applied once more to check whether such a case is coincidental or not. If a flagged entry deviates from the expected value one more significantly, it is evident that a specific input bit and a specific output bit are correlated, which is a major cryptographic weakness, so the matrix is considered to be non-random.

2.2 Linear Span Test

Nonlinearity is one of the basic design criteria for cryptographic primitives. In order to test block ciphers and hash functions for randomness based on nonlinearity, the outputs of a highly linearly dependent set of inputs are examined. For this purpose, linearly independent t plaintexts are chosen and an input set of size $m = 2^t$ is obtained by all linear combinations of these plaintexts. An $m \times m$ matrix is formed using the corresponding ciphertexts and the rank of this matrix is compared to the rank of a random binary matrix. After determining the rank of the output matrix, the corresponding bin value, which is initially set to 0, is incremented by one. After the test is repeated as many times as possible, the resulting bin values are put through a χ^2 Goodness of Fit Test with the probabilities given in Table 2 to produce the p-value. A p-value less than 0.01 is considered to indicate a non-random mapping.

Table 2. Probabilities used in Linear Span Test ($m > 19$)

Rank	$\leq m-2$	$m-1$	m
Probability	0.133636	0.577576	0.288788

The computation of the probability of a random binary matrix to have rank R for arbitrary R is not straightforward. However, an $m \times m$ random binary matrix has either rank m or $m-1$ over 85% of the time, therefore this test is applied with only three bins for the χ^2 Goodness of Fit Test and the probabilities for $Pr(R = m)$ and $Pr(R = m - 1)$ cases are needed. In the case $Pr(R = m)$, all ciphertexts are linearly independent. There are $2^m - 1$ choices for the first plaintext, $2^m - 2$ choices for second plaintext, $\cdots$, $2^m - 2^{i-1}$ choices for i^{th} plaintext, $\cdots$, $2^m - 2^{m-1}$ choices for the last plaintext, therefore

$$Pr(R = m) = \frac{\prod_{i=1}^{m}(2^m - 2^{i-1})}{2^{m^2}}$$

is obtained.

In the case $Pr(R = m - 1)$, first $m - 1$ linearly independent ciphertexts are chosen similar to the first case. The last ciphertext should be chosen so that, it is linearly dependent with the previously selected set. If the linearly dependent ciphertext is the i^{th} one, there are 2^{i-1} choices, thus there are $1 + 2 + 2^2 + \cdots + 2^{m-1} = 2^m - 1$ choices for it, therefore

$$Pr(R = m - 1) = \frac{\prod_{i=1}^{m-1}(2^m - 2^{i-1})}{2^{m^2}} \cdot (2^m - 1)$$

is obtained.

2.3 Collision Test

Collision resistance is an important design criterion for hash functions, which means that it should be hard to find two messages with the same hash value, and the Collision Test is designed to evaluate the randomness based on collision resistance. The subject of this test is the number of collisions in specific bits of the output, which can be considered as near collision. In other words, an input set of size n is evaluated through f, and the number of collisions (C) in t bits of the output is evaluated.

The same method with Knuth is used to calculate the probability that c collisions occur when n distinct random inputs are mapped into an output set of size $m = 2^t$ in [5]. The probability of c collisions occur is given as,

$$Pr(C = c) = \frac{m(m-1)\ldots(m-n+c+1)}{m^n} \left\{ \begin{matrix} n \\ n-c \end{matrix} \right\},$$

where $\left\{ \begin{matrix} n \\ n-c \end{matrix} \right\}$ is the Stirling number of the second kind. This probability is used to obtain the results given in Table 3. Here, as it affects directly the running time of the test, the selection of the parameter n should be done carefully. If it is chosen to be too large, it may not be possible to repeat the testing steps enough times to be able to detect subtle evidences of non-randomness. Therefore, 2^{12} and 2^{14} is chosen for the values of n in the calculations to have reasonable running times for the tests. The parameter m is chosen depending on the parameter n such that the probabilities are close enough to each other to be used in a 5-bin χ^2 Goodness of Fit Test.

Table 3. Ranges and probabilities of Collision Test for 16 and 20 bits

	$n = 2^{12}$, $m = 2^{16}$		$n = 2^{14}$, $m = 2^{20}$	
Bin	Range	Probability	Range	Probability
1	0-116	0.206246	0-117	0.190231
2	117-122	0.194005	118-124	0.215008
3	123-128	0.219834	125-130	0.211585
4	129-134	0.183968	131-137	0.202689
5	135-4096	0.195947	138-16384	0.180487

The test is applied as follows: first, a random input is taken and then an input set of size 2^{12} (or 2^{14}) is formed by assigning all possible values to its first 12-bits (or 14-bits respectively). After the outputs of these inputs are computed through f, number of collisions is obtained with the help of an array or a hash table. Afterwards, the corresponding bin value, is incremented. Above steps are repeated for 2^{20} random inputs and the resulting bin values are evaluated through a χ^2 Goodness of Fit Test with the expected values derived from Table 3 to produce the p-value. A p-value less than 0.01 is considered to indicate a non-random mapping.

2.4 Coverage Test

The Coverage Test evaluates a given function f through examining the size of the output set (coverage) formed from a subset of its domain. For a random mapping, the output set size is expected to be about 63% of the input set. Block ciphers loaded with a fixed plaintext and hash functions are one-way functions and required to behave like a random mapping. In the case of block ciphers, if the key is fixed, the function f becomes a random permutation and this case should be carefully investigated before moving on to the details of the test.

For a random permutation, the coverage is equal to the size of the input set, when all output bits are considered. For example, assume that the function f under test has block size n. Then, the coverage of an input set formed by 2^l distinct random values is obviously 2^l if all output bits are taken as reference.

But, if $n - k$ output bits are considered when forming the output set, the maximum possible number of hits that an $(n-k)$-bit output can have is 2^k. Therefore, whenever $l > k$, f is expected to behave like a random mapping.

Hence, when applying this test, functions used in both block ciphers and hash functions are expected to behave like a random mapping, which implies that the expected coverage is about 63% of the size of the input set.

The calculations for the intervals and their probabilities are the same with the Coverage Test proposed by Turan et al. in [4] for stream ciphers. Let A_k be the number of mappings from an n-element set to an n-element set. Here, A_k is defined recursively as

$$A_k = \binom{n}{k}\left[k^n - \sum_{i=1}^{k-1}\binom{k}{i}\frac{A_{k-i}}{\binom{n}{k-i}}\right] .$$

Therefore, the probability of the coverage being k is

$$Pr(C = k) = \frac{A_k}{n^n} ,$$

for $k = 1, 2, \ldots, n$. The probabilities and intervals for the bins to be used for the test are given in Table 4. This table is slightly different than the one given in [4], since a typo is spotted and corrected when verifying the results given in that paper.

Table 4. Ranges and probabilities of Coverage Test for 12 and 14 bits

	12 - bits		14 - bits	
Bin	Range	Probability	Range	Probability
1	1-2572	0.199176	1-10323	0.201674
2	2573-2584	0.204681	10324-10346	0.195976
3	2585-2594	0.197862	10347-10367	0.207530
4	2595-2606	0.203232	10368-10390	0.195266
5	2607-4096	0.195049	10391-16384	0.199554

The test is applied as follows: first, a random input is taken and then an input set of size 2^{12} (or 2^{14}) is formed by assigning all possible values to its first 12-bits (or 14-bits respectively). After applying f to this input set, the coverage is computed and the corresponding bin value is incremented.

The resulting bin values are evaluated through a χ^2 Goodness of Fit Test with the expected values derived from Table 4 to produce the p-value. A p-value less than 0.01 is considered to indicate a non-random mapping.

3 Application

In this section, application of the cryptographic randomness tests defined in Section 2 to the AES finalist algorithms and their results are given. Brief descriptions are followed by the testing results of each algorithm. Finally, a comparison with the previous work on the subject is stated.

3.1 MARS

The block cipher MARS[6] uses three main function when encrypting a block of plaintext. First, an unkeyed mixing operation called the forward mixing is applied following a key whitening operation. Then, the keyed transformation called the cryptographic core is applied to the state. Finally, another unkeyed mixing called the backward mixing is applied and the ciphertext is obtained after another key whitening operation. In this work, as it is desired to test the cryptographic randomness properties of the selected block ciphers, only the cryptographic core of the algorithm is tested. In other words, the forward and backward mixing operations are excluded when applying the tests defined in Section 2.

As Table 5 suggests, the cryptographic core of the algorithm satisfies the SAC property after 6 rounds. Since the cryptographic core has a type-3 Feistel network structure, this result is not unexpected. On the other hand, the suggested number of rounds for the cryptographic core is 16, which seems like a safe enough security margin.

Table 5. Number of rounds which MARS achieve randomness

SAC	Linear Span	Collision Test		Coverage Test	
-	-	16	20	12	14
6	2	3	3	3	3

3.2 RC6

RC6[7] is a 128 bit block cipher which uses 128,192 or 256 bit key sizes. The algorithm can be parametrized for other word and key sizes. The encryption process stars with a key whitening. Then, the round function is applied 20 times. The round function consists of modular multiplication, modular addition, XOR and rotations operations. 20 rounds is followed by another key whitening which concludes the encryption process.

The individual test results show that at least 5 rounds of the round function is enough to achieve randomness. The individual test results are given in Table 6.

Table 6. Number of rounds which RC6 achieve randomness

SAC	Linear Span	Collision Test		Coverage Test	
-	-	16	20	12	14
5	2	5	5	5	5

3.3 Rijndael

Rijndael[8] consists of four main operations: SubBytes, ShiftRows, MixColumns and AddRoundkey. The SubBytes operation is the confusion step, which uses an 8 × 8 s-box with very good cryptographic properties. The ShiftRows and MixColumns steps are mainly for satisfying the diffusion property. Finally, AddRoundkey is simply a key XOR at the end of each round. The MixColumn operation is skipped in the final round of encryption.

The results given in Table 7 suggests that randomness is achieved after 3 rounds. Moreover, the SAC property is satisfied after 4 rounds when the MixColumns operation is skipped in the last (fourth) round.

Table 7. Number of rounds which Rijndael achieve randomness

SAC	Linear Span	Collision Test		Coverage Test	
-	-	16	20	12	14
4	2	3	3	3	3

3.4 Serpent

Serpent[9] is an SP-Network using 32 rounds of successive substitution and permutation layers. Substitution layer consists of 32 4 × 4 *s*-boxes with good cryptographic properties. Permutation layer consists of a linear transformation using shift, rotation and XOR operations.

Serpent achieves randomness after 4 out of 32 rounds, which is a large enough security margin for block ciphers. The results for the individual tests are given in Table 8.

Table 8. Number of rounds which Serpent achieve randomness

SAC	Linear Span	Collision Test		Coverage Test	
-	-	16	20	12	14
4	2	4	4	4	4

3.5 Twofish

Twofish[10] is a 128-bit block cipher that can handle variable-length key up to 256 bits. The cipher is a 16-round Feistel network that uses key-dependent 8×8 s-boxes, a 4×4 maximum distance separable matrix, a pseudo-Hadamard transform and rotations.

Due to the fact that the round function of the Twofish algorithm has a Feistel network structure, only the even number of its rounds are tested. The results given in Table 9 suggest that the algorithm produces random outputs after 4 out of 16 rounds of its round function.

Table 9. Number of rounds which Twofish achieve randomness

SAC	Linear Span	Collision Test		Coverage Test	
-	-	16	20	12	14
4	2	4	4	4	4

3.6 Comparison with the previous work

During the AES selection process, J. Soto proposed a method to test block ciphers for randomness using the NIST Test Suite[2]. However, as the tests defined in that suite are more suitable to test long sequences, the block ciphers are considered as PRNGs and the outputs obtained from various input types are concatenated to form long sequences. After the sequences are generated, the statistical randomness tests defined in the NIST Test Suite are applied to these sequences. As a result for this testing process, a total of 189 p-values are produced from 16 tests for each input type.

Contrary to the above mentioned work on the subject, the tests proposed in this work are defined solely for the purpose of testing the cryptographic properties of the algorithms. Four tests are applied to the algorithms and a total of six p-values are produced. Although the number of p-values produced is relatively small, the results obtained from the cryptographic randomness tests are more precise than the results of the above mentioned work (see Table 10).

Table 10. Combined table stating the number of rounds which the algorithms achieve randomness

	SAC	Linear Span	Collision Test		Coverage Test		Previous Work[11]
Algorithms	-	-	16	20	12	14	-
MARS	6	2	3	3	3	3	4
RC6	5	2	5	5	5	5	4
Rijndael	4	2	3	3	3	3	3
Serpent	4	2	4	4	4	4	4
Twofish	4	2	4	4	4	4	2

4 Conclusion and Future Work

In this work, a package of randomness tests is proposed which evaluates block ciphers and hash functions through investigating their certain cryptographic properties. Throughout the work, SAC Test and Coverage Test, which were previously proposed for stream ciphers, are adapted to test the round functions of block ciphers and compression functions of hash functions. Also, Collision Test, which was originally proposed by Knuth for testing sequences, is adapted to test algorithms. Moreover, Linear Span Test is proposed to evaluate the ability of an algorithm to eliminate linear dependency of its input set. Afterwards, the package is applied to AES finalists and it is observed that the number of rounds where the randomness is achieved, is more precise for MARS, RC6 and Rijndael.

As it can be derived from the results given in Table 10, SAC Test, Collision Test and Coverage Test dominate the results, since Linear Span Test results show that most of the algorithms that are tested achieve randomness with a few rounds. But the test is included in the work, as hiding the linear dependency of its input is a nice and primitive property to have for a block cipher or a hash function.

Finally, as a future work, more cryptographic randomness tests can be defined to evaluate different properties expected from the round function of a block cipher or the compression function of a hash function. Also, the package can be applied to SHA-3 candidate algorithms.

References

1. A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, and S. Vo, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," 2001.
2. J. Soto, "Randomness testing of the AES candidate algorithms," 1999.

3. A.F. Webster and S.E. Tavares, “On the design of s-boxes,” Lecture notes in computer sciences; 218 on Advances in cryptology—CRYPTO 85, New York, NY, USA, pp.523–534, Springer-Verlag New York, Inc., 1986.
4. M.S. Turan, Ç. Çalık, N.B. Saran, and A. Doğanaksoy, “New distinguishers based on random mappings against stream ciphers,” SETA, ed. S.W. Golomb, M.G. Parker, A. Pott, and A. Winterhof, Lecture Notes in Computer Science, vol.5203, pp.30–41, Springer, 2008.
5. D.E. Knuth, Seminumerical Algorithms, The Art of Computer Programming, vol.2, Addison-Wesley, 1981.
6. C. Burwick, D. Coppersmith, E. D’Avignon, R. Gennaro, S. Halevi, C. Jutla, S.M.M. Jr, L. O’Connor, M. Peyravian, J. Luke, O.M. Peyravian, D. Stafford, and N. Zunic, “Mars - a candidate cipher for aes,” NIST AES Proposal, 1999.
7. B. Cipher, R.L. Rivest, M.J.B. Robshaw, Y. Yin, and R. Sidney, “The rc6 block cipher,” 1998.
8. J. Daemen and V. Rijmen, The Design of Rijndael: AES - The Advanced Encryption Standard, Springer, 2002.
9. E. Biham, R. Anderson, and L. Knudsen, “Serpent: A new block cipher proposal,” In Fast Software Encryption 98, pp.222–238, Springer-Verlag, 1998.
10. B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, and N. Ferguson, “Twofish: A 128-bit block cipher,” in First Advanced Encryption Standard (AES) Conference, 1998.
11. J. Soto and L. Bassham, “Randomness testing of the advanced encryption standard finalist candidates,” NIST IR 6483, National Institute of Standards and Technology, 1999.