Social Economic Debates
Volume 5, Issue 2, 2016
ISSN 2360-1973; ISSN-L 2248-3837

STATISTICAL ASSESSMENT OF BINARY SEQUENCES GENERATED BY CRYPTOGRAPHIC ALGORITHMS

Ioana Roxana Dragomir, PhD student
University Politehnica of Bucharest, Romania
Email: milita_roxana@yahoo.com

Abstract: *Along with the accelerated expanding/development of IT networks, there is an increase in the amount of information stocked, transmitted and processed in these networks, from both, public and private areas. The risk of losing such data may have a negative impact not only on people`s lives but on organizations` activity, particularly the governmental ones. That is why establishing actions intended to provide information security is an objective of high priority for military organizations and not only.*

For this category of information the security is provided by cryptographic algorithms which, in order to grant a high protection, must meet the principles identified by Shannon (1949): randomness, confusion and diffusion. This paper presents a method to verify the randomness by using a minimum suite of tests.

Keywords*: randomness; hypothesis testing; suite of tests; cryptographic algorithm; Advanced Encryption Standard.*

JEL Classification: C12, C61

1. Introduction

In the context of accentuated use of electronic environment for storage and transfer of information the development of information protection system, based on new techniques and technologies which provide security to a degree as high as possible, represent a major research field. Currently different encryption algorithms and techniques are used more and more frequently to provide information security. The problem with these encryption algorithms is that they should be, at least in theory, "unbreakable" to ensure the perfect protection. Therefore, before use these algorithms are tested by means of different methods to provide the desired security level.

This paper intends to submit a minimum number of tests with the help of which we can decide whether a given string may or may not be considered random, oriented at minimal logic level (bit). The tests are implemented in testing the randomness of the random and pseudo-random number generators, stream algorithms and even block algorithms, by using an adequate mode of operation (Dworkin, M., 2001). The paper presents a continuation of studying and developing testing tools for cryptographic primitives (Marin M., 2009), and for components of these ones (Dragomir, R. et altera, 2012, 2016). Random behavior is attested by a series of statistical tests applied on samples of sizes resulting from a large number of experiences. In scientifically literature there are a number of statistical tests suite used to check upon the randomness of data, such as: NIST (Rukhin, et altera A, 2000), DIEHARD (Marsaglia, G), but they are complex and execution of all these tests would mean long time. Therefore, a minimum suite of tests would solve the problem when a decision should be taken quickly. A string of random numbers is a set of real numbers "randomly chosen". Obviously, such a definition is unsatisfactory. The concept of random number only makes sense associated with the appearance of

an event within a certain experiences. Random numbers can be generated within the experiences and they cannot be determined a priori; at most, it may include only certain limits within which they can be contained.

In cryptography, the use of strings of random number and pseudorandom is indispensable. Therefore, in order to take a quick decision regarding randomness of a sequence suite of tests (Menezes et. altera 1997), was improved and implemented, certain tests were modified and one more test was added to the suite.

2. Methodology. Test description

The application comprises a set of six statistic tests have the role to verify the randomness of the binary sequences created by means of hardware and software generators of random and pseudorandom numbers.

Let $s = s_0, s_1, s_2, \ldots, s_{n-1}$ be a binary sequence having a length of n. The application submits to tests 1000 sequences - previously generated, each sequence having a length of 1.048.576 bits. For the significance level, three alpha values were taken into account: $\alpha_1=0.10$, $\alpha_2=0.05$, $\alpha_3=0.01$. The test that to be discussed are:

- Frequency of occurrence –Monobit test,
- Two-bit test,
- Poker test,
- Constant Runs test,
- Hamming Weight test and
- Autocorrelation test

These six statistical tests suite represents the minimum conditions for deciding if a sequence is random or not. If the sequence does not pass one of these tests no longer makes sense his assessment with more complex statistical tests.

2.1 Frequency of occurrence (Monobit) test.

The purpose of this test is to determine whether the number of 0's and 1's in s are approximately the same, as would be expected for a random sequence. This test evaluates the accuracy with which the report of one approaching ½ value so that the number of one and zero in a string should be the same. All subsequent tests depend on the promotion of this test.

The statistic used is

$$X_1 = \frac{(n_0 - n_1)^2}{n} \quad (1)$$

And the calculated value is compared with the chi-squared distribution with 1 degree of freedom for different levels of significance (see Table 1).

Level of Significance α / 2	CHI-SQUARE VALUE		Degrees of freedom
	min	max	
0.050	0.00393	3.84146	

0.025	0.00098	5.02389	1
0.005	0.00004	7.87944	

Table 1: Chi-Square Value for Frequency test

2.2 Serial test (two-bit test).

The objective of this test is to calculate the frequency of occurrence of each form of 2 bits in the entire string. The purpose of this test is to determine whether the number of occurrences of the four shapes (00, 01, 10, 11) of two bits is roughly the same number expected for a random string. Random strings are uniform, so every form 2 bit has the same probability of occurrence.

Let n_0, n_1 denote the number of *0*'s and *1*'s in *s*, respectively, and let n_{00}, n_{01}, n_{10}, n_{11} denote the number of occurrences of *00, 01, 10, 11* in *s*, respectively. Note that $n_{00} + n_{01} + n_{10} + n_{11} = (n - 1)$ since the subsequences are allowed to overlap. The statistic used is:

$$X_2 = \frac{4}{n-1}\left(n_{00}^2 + n_{01}^2 + n_{10}^2 + n_{11}^2\right) - \frac{2}{n}\left(n_0^2 + n_1^2\right) + 1 \quad (2)$$

which approximately follows a chi-squared distribution with 2 degrees of freedom for different levels of significance (see Table 2).

Level of Significance α / 2	**CHI-SQUARE VALUE**		**Degrees of freedom**
	min	max	
0.050	0.10259	5.99146	2
0.025	0.05064	7.37776	
0.005	0.01003	10.59660	

Table 2: Chi-Square Value for Serial test

2.3 Poker test

This test is used to test the randomness of a 4-bit nibble patterns in the capture file. The Poker Test is looking at four successive nibbles and place each four nibble pattern into one of four categories:

- r = 1 - One distinct value (All nibbles have the same value);
- r = 2 - Two distinct values;
- r = 3 - Three distinct values;
- r = 4 - Four distinct values (All nibbles have different values)

To calculate the X_3 statistic it is necessary to calculate the probability of each category.

- *Ps* = probability of observation *s*
- *k* = number of nibbles in each pattern category = 4
- *d* = number of ways to choose the four nibbles in the k window = 16

The probability of observation s will be given by formula 3:

$$P_S = \frac{d(d-1)\ldots(d-r+1)}{d^k}\left\{\frac{k}{r}\right\} \quad (3)$$

The *k* over *r* ration into brackets of formula 3 represents a mathematical operation known as Sterling Numbers of the second kind.

After the *p-values* for all categories are computed and the observations have been counted by category, the X_3 statistic can be calculated,

$$X_3 = \left[\sum_{i=0}^{3} \frac{(Y_i)^2}{NP_i} \right] - N \quad , \tag{4}$$

were:

K = number of categories = 4
N = number of observations
Y_s= number of observations in category s
X_3 = calculated statistic for the Poker test

The X_3 statistic is compared to the chi-squared distribution with 3 degrees of freedom. Possible values for the significance levels, and the corresponding chi-square values are in the Table 3.

Level of Significance α / 2	**CHI-SQUARE VALUE**		**Degrees of freedom**
	min	max	
0.050	0.351846	7.81473	3
0.025	0.215795	9.3484	
0.005	0.0717218	12.8382	

Table 3: Chi-Square Value for Poker test

2.4 Constant Runs test.

The objective of this test is to calculate the total number of uninterrupted sequences of bits equal to zero or one. An uninterrupted sequence of identical bits of length k is a substring consisting of k bits of the same value (zero or one) and which is bounded by a bit of different value. The purpose of the runs test is to determine whether the number of runs (of either zeros or ones) of various lengths in the sequence s is as expected for a random sequence. The expected number of gaps (or blocks) of length *i* in a random sequence of length *n* is given by formula (5).

$$e_i = \frac{(n-i+3)}{2^{i+2}}. \tag{5}$$

Let *k* be equal to the largest integer *I* for which $e_i \geq 5$. Let B_i, G_i be the number of blocks and gaps, respectively, of length *I* in *s* for each *i,* $1 \leq i \leq k$. The statistic used is given by formula (6).

$$X_4 = \sum_{i=1}^{k} \frac{(B_i - e_i)^2}{e_i} + \sum_{i=1}^{k} \frac{(G_i - e_i)^2}{e_i} \tag{6}$$

which approximately follows a chi-squared distribution with 2k – 2 degrees of freedom (see Table 4).

Level of Significance α / 2	CHI-SQUARE VALUE		Degrees of freedom
	min	max	
0.050	18,4927	43,7730	30
0.025	16,7908	46,9792	
0.005	13,7867	53,6720	

Table 4: Chi-Square Value for Constant Runs test

For our application, we have k = 15 and the next values of e_i (e_{16}<5) are given by Table 5:

e_1	131072.250000
e_2	65536.062500
e_3	32768.000000
e_4	16383.984375
e_5	8191.984375
e_6	4095.988281
e_7	2047.992188
e_8	1023.995117
e_9	511.997070
e_{10}	255.998291
e_{11}	127.999023
e_{12}	63.999451
e_{13}	31.999695
e_{14}	15.999832
e_{15}	7.999908
$\geq e_{16}$	7,999908

Table 5: The results for the expected number of blocks

2.5 Hamming Weight test

The Hamming Weights of the data in the capture file are counted for each possible weight, 0 to 32. The Hamming Weight is the number of ones in the 32 bit word. Although 33 different Hamming Weights are possible, the expected value of the very high and very low Hamming Weights are extremely low. A general rule of thumbs in statistics is that each category should have an expected value of at least five. To comply with this rule, Hamming Weight 0 through 7 are combined into a single category and Hamming Weight 25 through 32 are combined into a second category. With each Hamming Weight from 8 to 24 comprising its own category that results in a total of 19 categories for the Hamming Weight test.

- $K_1 = 1$, all words in which the number of bits "1" is less than 8
- $K_2 = 2$, all words in which the number of bits „1" is 8
- $K_3 = 3$ all words in which the number of bits „1" is 9
- .....
- $K_{18} = 18$ all words in which the number of bits „1" is 24

- K_{19} = *19* all words in which the number of bits "1" is larger than 24

The Hamming weight is the number of "1" bits in a 32-bit word. To calculate the X_5 statistic it is necessary to calculate the probability of each category.

- K = number of categories = 19
- P_s = probability of observation s
- N = number of observations
- Y_s = number of observations in category s
- N_b = number of bits in the sample word = 32
- q = probability of a bit being a one = 1/2
- p = probability of a bit being a zero = 1/2
- h = Hamming Weight
- P_h = probability of Hamming Weight h, calculated as formula (7).

$$P_h = \frac{32!}{h!(32-h)!}(p)^{32} \tag{7}$$

For Hamming Weights 8 through 24 P_h is equal to P_s. For the categories that combine several Hamming Weights the probabilities for each Hamming Weight are added together to form a probability for the category. X_5 = calculated statistic for the Hamming Weight test is given by formula (8).

$$X_5 = \left[\sum_{i=0}^{18} \frac{(Y_i)^2}{NP_i} \right] - N \tag{8}$$

The calculated X_5statistic is compared to the chi-square distribution with 18 degrees of freedom.Possible values for the significance levels, and the corresponding chi-square values are given in Table 6.

Level of Significance α / 2	**CHI-SQUARE VALUE**		**Degrees of freedom**
	min	max	
0.050	9.39046	28.8693	18
0.025	8.23075	31.5264	
0.005	6.26480	37.1565	

Table 6: Chi-Square Value for Hamming Weight test

2.5 Autocorrelation test

The purpose of this test is to check for correlations between the sequence s and (non-cyclic) shifted versions of it. Let d be a fixed integer (see formula 9).

$$1 \le d \le \left\lfloor \frac{n}{2} \right\rfloor. \tag{9}$$

The number of bits in s not equal to their d-shifts is given by formula (10).

$$A(d) = \sum_{i=0}^{n-d-1} s_i \oplus s_{i+d}, \tag{10}$$

Where the sign $\oplus$ denotes the XOR operator.

The autocorrelation statistic *A* is a binomial statistic with parameters *N*, *p* and *q*, where $N = n - d$; $p = \frac{1}{2}$; $q = 1 - p$.

The binomial statistic has mean μ and standard deviation σ, $\mu = N \cdot p$; $\sigma = \sqrt{N \cdot p \cdot q}$ .A good method for judging the significance of an autocorrelation score (A) is to transform *A* to *Z* values by Z transformation (see formula 11).

$$Z = \frac{A - \mu}{\sigma} \tag{11}$$

The new statistic *Z* has a normal distribution with mean μ = 0 and variance σ^2 = 1.So, Z becomes (see formula 12):

$$Z = \frac{A(d) - (n-d) \cdot p}{\sqrt{(n-d) \cdot p \cdot q}} = \frac{2 \cdot A(d) - (n-d)}{\sqrt{n-d}} \tag{12}$$

Hence, the statistic used is given into formula 13.

$$X_6 = \frac{2 \cdot \left(A(d) - \frac{n-d}{2} \right)}{\sqrt{n-d}} \tag{13}$$

which approximately follows an *N(0, 1)* distribution if $n - d \geq 10$. Since small values of *A(d)* are as unexpected as large values of *A(d)*, a two-sided test should be used. For our application, it was taken d = 8.

Level of Significance α / 2	**STANDARD NORMAL DISTRIBUTION VALUE**	
	MIN	MAX
0.050	-1.6449	1.6449
0.025	-1.9600	1.9600
0.005	-2.5758	2.5758

Table 7: Autocorrelation test theoretical values

3. Experiment and results

The objective of this experiment is to prove the usefulness of this suite of tests, which is why we have chosen for the experimental side the only standardized cryptographic block algorithm Advanced Encryption Standard [9].

Thus the 1000 samples subjected to testing were generated with AES-CTR, Counter (CTR) (Dworkin, M , 2001) is a mode of operation that allows a block algorithm to behave like a stream algorithm.

The six statistical tests are applied to 1000 samples each with a length of 1048576 bits. For it were developed two applications in C++ programming environment one to generate the samples to be tested and the other for running the tests. Statistics calculation was made in Wolfram Mathematica 6.00.

The results of this experiment are presented in the next table. The results are displayed for three significant threshold values:

AES-CTR						
Level of significance $\alpha/2$	**T1 FREQUENCY**	**T2 SERIAL**	**T3 POKER**	**T4 CONSTANT RUNS**	**T5 HAMMING WEIGHT**	**T6 AUTO CORRELATION**
0,05	875	907	883	885	929	952
0,025	984	985	984	946	992	968
0,005	1000	1000	992	992	992	1000

Table 8: Results of AES-CTR testing

Thus each column are the results of the six tests, each test is the number of samples that have stood the test of the 1000 tested. We can see how it was expected that the results of this algorithm are very good.

4. CONCLUSION

This paper presents a method with which we can quickly say if a sample is random or not. The statistical testing may represent the most important stage in the security assessment of an algorithm. Statistical testing does not mean the lack of need for cryptanalysis. The cryptographic algorithm represents the core of IT items security. That is why assessing them with reproducible results is essential. The statistical testing of a cryptographic primitive must be a quick process, so as to not increase the complexity of the cryptographic system in which it is implemented. The randomness property is verified by a suite of frequency of occurrence tests sufficient to decide if a sample is random or not. If a sample doesn't pass those tests no longer makes sense application of other more complex statistical tests.

For high security applications this minimal suite of tests can be considered a phase preceding a wider statistical testing

It is intended, in the near future, to study and to develop more of such instruments needed to evaluate a cryptographic algorithm.

5. REFERENCES

Daemon, J., Rijmen, V.,"The Design of Rijndael", Springer 2002, ISBN: 3-540-42580-2;
Dragomir I.R., Filip, D., Marin, M., "Boolean functions used in cryptology ", ISETC 2012, ISBN: 978-1-4673-1176-2, pp. 175-178; http://iasmina.cm.upt.ro/work/2012-ISETC/TOT.pdf;
Dragomir I.R., Lazar, M.,"Generating and testing the components of a block cipher", ECAI 2016 -International Conference – 8th Edition Electronics, Computers and Artificial Intelligence 30 June - 02 July, 2016, Ploiesti, ROMÂNIA International Journal of Computer Science and Security (IJCSS), Volume 6, 2012;
Dragomir, R., Marin, M., Rastoceanu, F., Roman, F. "Testing block cipher strength with diffusion method", The 18th International Conference the Knowledge Based Organization, 14-16 June 2012, ISBN: 1843-6722, pp. 218-222;
Dworkin, M., NIST Special Publication 800-38A 2001 Edition „Recommendation for Block Cipher Modes of Operation – Methods and Techniques";
Marin, M., Răstoceanu, F., Dragomir, R., "Statistical testing of random number generators", AFASES 2009 – Scientific research and education in the air force 20-22 May Brasov, ISBN: 978-973-8415-67-6;
Marsaglia, G., DIEHARD Statistical Tests , See http://stat.fsu.edu/~geo/diehard.html;

Menezes, A. J., van Oorschot, P. C. and Vanstone, S. A., "Handbook of Applied Cryptology", CRC Press, Inc., 1997, chap5.pdf.zip;
Rukhin, A., Suoto, J.,NechvatalJ. et. al, *"A Statistical Test Suite for the Validation of Random and Pseudo Random Number Generators for Cryptographic Applications",* NIST Special Publication 800-22, 2000;
Shannon, C. E, "Communication Theory of Secrecy Systems", Bell Systems Technology Journal, vol. 28, nr. 4, 1949, p. 656-715;