Statistical assessment - as a part of security assessment applied to a block cipher

Ioana Roxana DRAGOMIR (milita_roxana@yahoo.com)
University Politehnica of Bucharest, Romania
Marilena LAZĂR (mnvlazar@yahoo.com)
University Politehnica of Bucharest, Romania

ABSTRACT

The security provided by the block cipher algorithms is a top modern approached issue within the specific researches. The security assessment of a block algorithm involves the assessing and testing of components, the statistical testing of algorithm and last, but not least, the cryptanalysis. This paper intends to approach a part of this issue by presenting one of the security assessment stages, namely the component analysis, together with the statistical testing of the block cipher. The statistical testing may be considered as a first step in the security assessing of a block algorithm.

The statistical testing of randomness of the block cipher represents an essential phase, both, for the research-development process, and for the assessment process applied to the cryptographic primitives, taking into account that the block algorithms are used on a large scale in cryptographic applications. Assessing them, from a cryptographic point of view, is a highly complex task, which cannot be efficiently accomplished by formal methods. This paper presents several statistical methods of carrying out a security assessment on a block algorithm.

Keywords: *key schedule, statistical testing, cryptographic components, randomness, tests, cryptographic primitive*

JEL Classification: *C4*

INTRODUCTION

Cryptography is a small piece, but a very important one, from within a puzzle meant to provide security.

The process of testing the security provided by the block algorithms is a very complex and expensive one, due to the fact that the difficulty of testing increases along with the plaintext block size, and with the key length. Because, even for a value of 64 bits, it is not possible to carry out exact tests (today the values that are currently used are 128 bits, and 256 bits), statistical methods can be used for approaching the issue. Consequently, the result of the tests is a

probabilistic one; the successful completion of the tests represents a necessary but not sufficient condition („sine qua non" requirement, still insufficient) for the algorithm to be considered secure.

The testing process is performed in several stages. First, a general assessment - a black-box analysis type - is carried out, without taking into account the internal structure of the block algorithm; further on, if necessary, a complex particular white-box type analysis is conducted, which consists in verifying the mathematical pattern of the algorithm and in performing cryptanalysis. The black box analysis verifies the main properties of the block algorithm: randomness, diffusion and confusion.

Most researchers are characterized by the propensity to confuse statistical testing with security testing, the last one representing a much more complex process, which involves cryptanalysis and various attack scenarios.

Security depends not only on the complexity of the algorithm but also on the context this one is applied in, as for instance: the block type algorithms are used in offline encryption, yet, when they are applied within a certain cryptographic mode (CTR, CFB etc.) they behave as stream (ciphering) algorithms. Thus, a block type algorithm may be implemented in complex cryptographic systems created for handling information. To carry out cryptanalysis on such a cipher may mean to analyze the data traffic from which information about the key may be extracted, and, consequently, about the plaintext.

At present, there are some standardized testing tools such as the battery of tests NIST [2] which can be applied to stream ciphers as well as to block ciphers in a slightly modified manner[1]. The interest shown by the cryptographic (scientific) community in this very important part belonging to algorithm assessment is illustrated by the great amount of published papers [1], [6], [7], [9], [10].

STAGES IN TESTING A BLOCK TYPE ALGORITHM

The structure of the block algorithm

First, in the structure of the block cipher there are two stages. The first stage is the initialization stage (in which round keys are generated) followed by the second one the processing stage (in which data processing takes place, namely: ciphering/deciphering processes).

The stages implied by the statistical testing of the block algorithm

It is considered that the statistical evaluation of a block algorithm takes place in three stages:

- Analysis and assessment of the components which may be: Boolean functions, S-boxes, polynomials; the context in which these components are used is very important, namely during the initialization stage or during the processing stage.

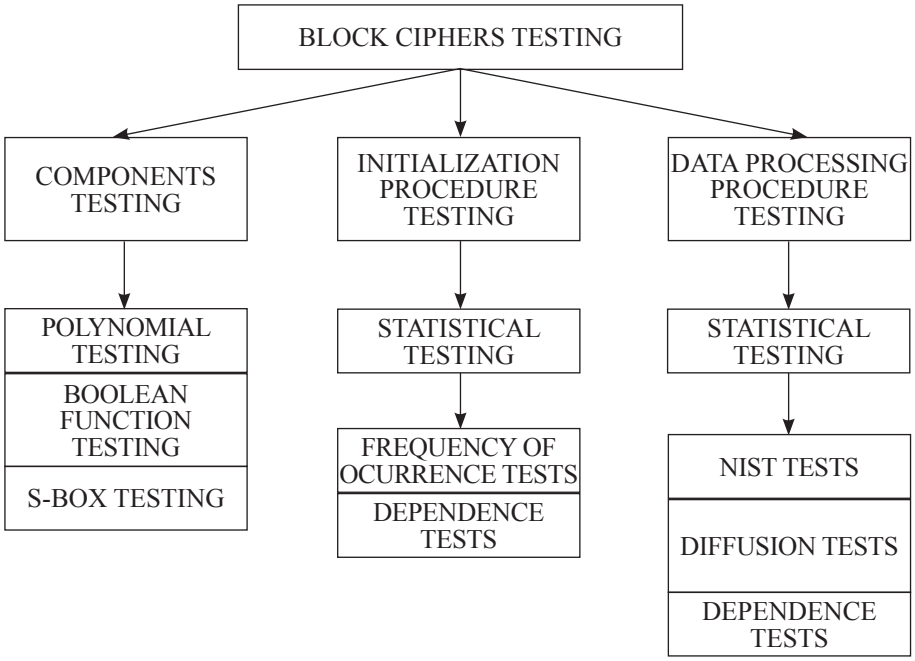
- Statistical assessment for which we implemented statistical tests known from specific publications, as adjusted to the initializing stage.

- Statistical assessment of the data processing stage, i.e. of the ciphering/deciphering process.

In the next schema we synthesize the stages of testing a symmetric algorithm:

STAGES IN TESTING A BLOCK ALGORITHM

Figure 1



The purpose of these experiments was to develop procedures for statistical testing of an algorithm. The accuracy of implementation(s) was verified with a standardized algorithm, such as AES.

In order to perform the tests, each of the three stages was threated individually. Within the first stage, the components were tested; in case of AES algorithm, it was dealt with one S-box built up of 8 Booleans functions

of eight variables. For the statistical assessment of the initialization stage, frequency of occurrence tests and dependence tests [7] were conducted, as described further on. Within the third stage i.e. the statistical assessment of the data processing stage, the NIST test battery [2] was applied on eight data types [1], diffusion tests [10], and dependence tests [7].

TEST DESCRIPTION

Assessment of the components

The assessment of the components represents an important stage, because from a cryptographic point of view, using weak components, may lead to a weak algorithm.

When the algorithm is submitted to statistical testing, it may be viewed as a black-box; when the components are submitted to testing, one must have access to the mathematical model of the algorithm, so as to be able to identify the components.

The testing of the Boolean functions was approached in a previous paper [5] where it was showed which are the properties of a robust, Boolean function, and how can they be generated. The Boolean functions are the smallest - but important - components upon which, the robustness of the algorithm depends. Several Boolean functions make up an S-box. The AES algorithm has, as a main component, a fixed size S-box, with the correct properties, used both, in the stage where round keys are generated, and in the data processing stage.

An application in C++ was developed, which is designed to compute the properties of the S-box [12]. The accuracy of these implemented properties was certified by means of the AES S-box.

Other very important category of components that can constitute a cipher is represented by the polynomials, which usually, in cryptography, should be primitives. Because working with high degree polynomials is a practice, an application in Wolfram Mathematica 6 was developed, by means of which it can be established if a polynomial is primitive or not, and if not, which are the factors.

Assessment of the initializing stage

The initialization stage consists in the round key generation. The purpose of key testing was to verify the Shannon features [11], confusion and diffusion. In cryptography, confusion and diffusion are the two properties of a cipher that were identified by Shannon in his article „Communication Theory of Secrecy Systems" published in 1949.

In a cipher with a good diffusion, changing the value of one input bit will change each output bit with a 50% probability (Avalanche Criterion).

Substitution (a plaintext symbol is replaced by another one) was identified as a basic mechanism of confusion (see S-box), but transpositioning (rearranging the symbols in a different order) is a diffusion technique, although there are some other mechanisms currently used in modern cryptography, such as linear transformations [8].

The confusion property is verified by frequency of occurrence tests, whereas the diffusion property is verified by dependence tests. With AES algorithm, the initialization stage is almost as complex as the data processing stage.

All these tests are presented by the authors in a paper which at the moment is being assessed in order to be published.

Assessment of the data processing stage

The data processing stage is the most important phase of the testing process. It is, in fact, the testing of the algorithm. Testing the randomness involves determining the statistics over a fragment of data binary string or over the entire data set. The results of statistical tests are compared with expected values which are established by using a random numbers generator with a flat distribution (practicality, when specific software is conducting the test: the p-value method is used to decide upon a test result).

If the data binary string passes a certain test it does not mean that the random numbers generator with flat distribution generates really random numbers, i.e. even distribution numbers; it only means that particular test cannot make the difference between the analyzed sequence and a really random binary string.

If the data binary string does not pass a certain test, then we can say that that test makes the difference between the analyzed binary string and a random data sequence.

The more tests are applied and passed by the analyzed binary string, the stronger becomes the confidence that this binary data string belongs to the random numbers range (set). In order to determine the randomness degree of a data sequence, a set of tests was used, containing tests specifically for randomness which are complementary (each test tracks aspects of statistical properties of the analyzed string, different from those verified by the previous tests). Examples of such tests are: the NIST battery of tests or Diehard battery of tests.

Randomness assessment was performed by means of NIST battery of tests. NIST battery of tests contains 16 statistical tests developed to verify the randomness of binary sequences (with arbitrary length) which were generated

for cryptographic purposes either by hardware or software generators of random numbers.

As these statistical tests generate a correct (an acceptable) set of statistic data, they may be used, just as well, in case of block type algorithms.

Testing methodology of symmetrical block type algorithms consists of two stages. Within the first stage, eight different types of test sequences are generated with the tested algorithm; during the second stage these data categories are analyzed with NIST battery of tests. The types of data categories are presented in the Table 1.

DATA CATEGORIES TYPES

Table 1

Types of data categories	Objective of the analysis	Length of the sequence (bits)
Plaintext Avalanche	Examine the responsiveness of the algorithm to changes made in the plaintext	1048576
Key Avalanche	Examine the responsiveness of the algorithm to changes made to a key	1048576
Plaintext/ Ciphertext Correlation	Study the correlation plaintext / ciphertext	1048576
Cipher Block Chaining Mode	Study ciphertext obtained by using encrypting in CBC mode	1048576
Low Density Plaintext	Study the algorithm in the situation that the plaintext used has a low density (almost all the bits are ,0')	1056896
Low Density Key	Study the algorithm in situation that a key with low density is used (almost all the bits are ,0')	1056896
High Density Plaintext	Study the algorithm in the situation that the plaintext used has a high density (almost all the bits are ,1')	1056896
High Density Key	Study the algorithm in situation that a key with high density is used (almost all the bits are ,1')	1056896

Statistical tests are mechanisms designed to support making quantitative decisions regarding a process, ex.: ciphering operation performed by a block type algorithm.

Because these types of data are mapped in a similar manner, the structure of only one type of data will be further described, namely the data related to the first category submitted to testing.

Plaintext Avalanche – used to examine the responsiveness of the algorithm to the modifications made in the plaintext. To this end, 300 binary sequences were analyzed (each sequence having 1048576 bits). The 300 sequences were obtained from a string created according to the following procedure:

• a number of 19200 random blocks of plaintext is generated PT_i *i=1,2,...,19200*, each having a length of l_{PT}= 128 bits;

• out of each PT_i block, 128 perturbed blocks are obtained PT_{ij}, *j=1,2,...128* by changing the value of the bit filling „j" position (*j=1,...,128*);

• each of the PT_{ij} blocks is encrypted with the algorithm tested in ECB mode, with a key length of 128 „0" bits, resulting the ciphered block CT_{ij};

• each of the C_{ij} blocks is summed up modulo 2 with the random block of plaintext PT_i, thus, D_{ij} blocks are obtained.

The result of all these procedures is a bit string with a length of l_{PT}*i*j = 128*19200*128 = 314.572.800 bits (39321600 bytes) which represents the 300 sequences with a length of 1048576 bits (300*1048576 =314.572.800).

NIST battery of tests has 16 tests and 189 subtests. Each data category was submitted to NIST battery of tests.

Comments on the results of the statistical testing. In order to interpret the results of the statistical tests, NIST endorsed two approaches:

1) examining the proportion of sequences which passed a statistical test for which the null hypothesis was not rejected and
2) checking the distribution of ***P-value****(s)* - in order to verify the uniformity.

The general condition to validate the randomness of a random number generator is to perform both assessment processes (proportion & uniformity). The ***p*** proportion of the sequences which pass a statistical test:

• the sequence passes if $\boldsymbol{p_{test}} > 0.01$;

• the array of acceptable proportions is established/identified by the trust range which is defined as follows:

$$\beta \pm 3 * \sqrt{\frac{\beta(1-\beta)}{n}}, n = 300, \alpha = 0{,}01, \beta = 0{,}99 \quad (1)$$

• the cut-off or the critical value is 0,972767.

The distribution of ***P-value*** - to verify the uniformity :

• the range [0,1] is divided into 10 sub-ranges, then the amount of *P-value(s)* existing in a sub-range is computed;

• the uniformity is emphasized by applying the conformity / compliance test:

$$\chi^2 = \sum_{i=1}^{10} \frac{\left(F_i - \frac{n}{10}\right)^2}{\frac{n}{10}} \quad (2)$$

where: $\boldsymbol{F_i}$ **–** the number of ***P-value****(s)* present in sub-range *i*;

n – number of sequences and,

trust level ***P-value*** > *0,0001*.

Assessment of the diffusion principle. Diffusion is one of the basic principles that makes a block cipher algorithm secure. Diffusion means that each bit of the ciphertext depends on each bit of the plaintext and of the key. Any minor change (modification) of the plaintext or of the key are causing a major and random change of the ciphertext. Otherwise, the cryptanalysts may notice or infer relations between these elements, all this leading to a significant diminish of the area where keys may be looked up. In this paper, we present a method to test block ciphers [10].

To this stage, a C++ application was developed, which automatically tests the diffusion property in block ciphers. In order to verify how well the diffusion property was implemented, the testing was performed on a known algorithm: AES.

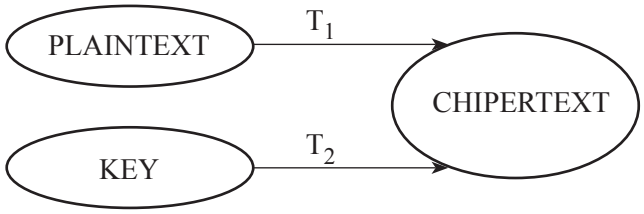
The diffusion principle consists in dissipating the statistical properties of the plaintext in the ciphertext. In fact, this means that preserving the same key, the change of one single bit (any bit) of the input block results in major changes in the output blocks. Theoretically, each output bit is changed with a probability of ½, in other words, each bit of the input block affects many of the output block bits (over 50%). In a similar way, while the plaintext block is preserved unaltered, changing one single bit (any bit) from the key causes an avalanche effect over the output block (50% of the ciphertext bits are affected).

The diffusion testing involves two subtests which are examining the effect that each plaintext bit has over the obtained ciphertext block (T1) and the effect that each key bit has over the ciphertext block obtained from a plaintext block (T2).

For the first test (T1), the content of the random plaintext block is reserved unaltered, while for the second test (T2), the randomly generated key is preserved unchanged.

DIFUSSION SUBTESTS

Figure 2



The examination of the diffusion property is performed by a procedure which is repeated N times; in the end, the conformity / compliance test, the chi squared, having the symbol c2, is applied. To this end, the resulted

observations are arranged into 5 groups (categories) of observations, with the same probability. Then, the frequency of occurrences within each category is compared to the expected values, in accordance with the chi squared statistic.

Note: The implementation of conformity/compliance test chi squared, requires N to be a pretty large number (N>100); when implementing the diffusion test, the value N=1024 was agreed.

For a secure cipher, the distribution of observations is binominal, with the parameters n and the probability p=½, meaning that the probability coming out of the total *n* number of bits, exactly *i* bits, should be equal to „1", and this is determined by the following formula:

$$P_i = C_n^i \cdot p^i (1-p)^{n-i} = C_n^i \cdot \left(\frac{1}{2}\right)^n . \quad (3)$$

For N = 1024, the boundaries of the 5 categories were chosen so as to have the same probability (»1/5)

0 - 498, (2) 499 - 507, (3) 508 - 516, (4) 517 – 525,

(5) 526 – 1024.

The probability for each category was determined based on Wolfram Mathematica 6 software, by applying the following computations:

$$P_1 = \sum_{i=0}^{498} \frac{C_{1024}^i}{2^{1024}} = 0.19941 = \sum_{i=526}^{1024} \frac{C_{1024}^i}{2^{1024}} = P_5$$

$$P_2 = \sum_{i=499}^{507} \frac{C_{1024}^i}{2^{1024}} = 0.18986 = \sum_{i=517}^{525} \frac{C_{1024}^i}{2^{1024}} = P_4$$

$$P_3 = \sum_{i=508}^{516} \frac{C_{1024}^i}{2^{1024}} = 0.22146$$

The statistic calculus is as formula 4:

$$S = \sum_{i=1}^{5} \frac{(f_i - m_i)^2}{m_i} \quad (4)$$

where f_i is the identified empirical frequency of occurrences, and $m_i = N_{obs} x P_i$ is the theoretical frequency of occurrences, so as according to formula 5.

$$\sum_{i=1}^{5} f_i = \sum_{i=1}^{5} m_i = N_{obs} \quad (5)$$

The critical values of the chi squared test are in this case 0,297 (the lower limit) and 13,277 (upper limit); these values rank within a certain

range depending on the number of freedom degrees, and on the chosen significance limit / cut-off value: the number of freedom degrees is smaller by a mathematical unit than the number of observation categories, i.e. 4, and the limit is 0,01, for the required significance.

Assessment of statistical dependence property - was the topic of a past paper of the authors [7] in which several important properties were analyzed and implemented – completeness property, avalanche, and strict avalanche. These tests play an important role both in the developing stage (they help determining the optimum number of rounds), and in the assessing stage.

The dependence tests are specific to the block type algorithms, and they verify the meeting of criteria such as complete transformation, avalanche, and strict avalanche, over a number of 10000 samples of key / plaintext, by modifying one plaintext bit at a time (bit by bit).

The next concepts are necessary to establish the four dependence criteria:

- NBITS = the average number of output bits altered consequently to a modification of one input;
- TDC = the degree of completeness;
- TDA = the degree of avalanche effect;
- TDSA = the degree of strict avalanche effect.

The mathematical description of the tests:

For a vector $x = (x_1, ..., x_n) \in \{0,1\}^n$, the vector $x^{(i)} \in \{0,1\}^n$ denotes the vector obtained by complementing the bit i belonging to x (i=1, ..., n).

A function $f : \{0,1\}^n \to \{0,1\}^m$ with n input bits, and m output bits is considered to be complete if each output bit depends on each input bit, namely: $\forall i = 1, ..., n , \forall j = 1, ..., m , \exists x \in \{0,1\}^n$ so as to $(f(x^{(i)}))_j \neq (f(x))_j$.

A function $f : \{0,1\}^n \to \{0,1\}^m$ exhibits the avalanche effect AVAL (Avalanche Criterion) if one single input bit is complemented and, consequently, half of the output bits (on average) are modified, that is: $\frac{1}{2^n} \sum_{x \in \{0,1\}^n} wt(f(x^{(i)}) - f(x)) = \frac{m}{2}$, , for all i = 1, ..., n.

A function $f : \{0,1\}^n \to \{0,1\}^m$ satisfies the Strict Avalanche Criterion (SAC) if one single input bit is complemented, and each output bits is modified with a probability of 1/2 , namely:

$$\forall i = 1,...,n, \forall j = 1,...,m, \Pr\left((f(x^{(i)}))_j \neq (f(x))_j\right) = \frac{1}{2}.$$

The dependence matrix of a function $f: \{0,1\}^n \to \{0,1\}^m$ is a matrix $A_{n \times m}$, whose elements a_{ij} denote the number of inputs for which the complementation of the input bit i causes the change of the output bit j,

namely: $a_{ij} = \#\{x \in \{0,1\}^n \mid (f(x^{(i)}))_j \neq (f(x))_j\}$ for i = 1, ..., n and j = 1, ..., m.

The distance matrix of a function $f : \{0,1\}^n \rightarrow \{0,1\}^m$ is a matrix $B_{n\times(m+1),}$ whose elements b_{ij} denote the number of inputs for which the complementation of the input bit i causes the change of j output bits, namely: $b_{ij} = \#\{x \in \{0,1\}^n \mid w(f(x^{(i)}) - f(x)) = j\}$ for i = 1, ..., n and j = 0, ..., m.

The dependence matrix is defined as follows:
$a_{ij} = \#\{x \in X \mid (f(x^{(i)}))_j \neq (f(x))_j\}$ for i = 1, ..., n and j = 1, ..., m. The distance matrix is defined as follows: $b_{ij} = \#\{x \in X \mid w(f(x^{(i)}) - f(x)) = j\}$ for i = 1, ..., n , j = 0, ..., m, where X is a subset of $\{0,1\}^n$ „convenient" randomly chosen. Let us assume that, the dependence matrix A has to be computed and the distance matrix B of a f: $\{0,1\}^n \rightarrow \{0,1\}^m$ for a set of inputs X, where X is $\{0,1\}^n$ or a random subset of $\{0,1\}^n$.

The completeness degree of f function is defined as in formula 6:

$$TDC = 1 - \frac{\#\{(i,j) \mid a_{ij} = 0\}}{n \cdot m} \tag{6}$$

The degree of avalanche effect of f function is showed by formula 7.

$$TDA = 1 - \frac{\sum_{i=1}^{n} \left| \frac{1}{\#x} \sum_{j=1}^{m} 2 \cdot j \cdot b_{ij} - m \right|}{n \cdot m} \tag{7}$$

The degree of strict avalanche effect of f function is given by formula 8.

$$TDSA = 1 - \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} \left| \frac{2 \cdot a_{ij}}{\#x} - 1 \right|}{n \cdot m} \tag{8}$$

In order for the f function to have valid values for the degree of completeness, degree of avalanche, degree of strict avalanche, the numbers of TDC, TDA and TDSA must meet the requirement: TDC = 1, TDA ≈ 1, and TDSA ≈ 1.

TESTS RESULTS

The testing of the components

The AES S-box has strong cryptographic properties, and can be split into eight Boolean functions which also have these kind of strong properties. We shall compute the properties of Boolean functions as shown in article [5] and the properties of the S-box.

S-BOX AND BOOLEAN FUNCTIONS PROPERTIES

Table 2

AES properties			
S-box properties		**Boolean functions properties of S-box**	
Differential potential	4/256	Nr. var.	8
Robustness	0.9843	Balancedness	yes
Differential Branch Number	2	Nonlinearity	112
Linear Branch Number	3	Correlation Immunity	0
Linear potential	16/128	Algebraic Degree	7
Nonlinearity	112	Algebraic Immunity	4
Algebraic Degree	7	Auto-correlation	32

The testing of the processing stage using NIST battery of tests

The method was implemented in Visual C++ 6.0 and the developed application allows testing any block type algorithm implemented in C++.

NIST TESTS RESULTS

Table 3

Randomness Evaluation				
Data type	**Proportion**	**Uniformity**	**Passed**	**Decission (%)**
Plaintext avalanche	187 of 189	189 of 189	187 of 189	98.94
Key avalanche	189 of 189	189 of 189	189 of 189	100
Plaintext/ Ciphertext correlation	188 of 189	189 of 189	188 of 189	99.47
Cipher block chaining mode	189 of 189	189 of 189	189 of 189	100
Low density plaintext	186 of 189	189 of 189	186 of 189	98.41
Low density key	188 of 189	189 of 189	188 of 189	99.47
High density plaintext	188 of 189	189 of 189	188 of 189	98.47
High density key	189 of 189	189 of 189	1879 of 189	100

The implementation must involve three functions intended to provide the interface, the key setting function, the encrypting function and the decrypting function.

First, the eight types of data were generated by means of AES algorithm and then, each of the eight types was submitted to testing with NIST battery of tests. The results are presented in the table III. We specify that the confidence level was 0,01 (1%).

Assessment of the diffusion property. In order to give an example, two diffusion subtests applied to AES algorithm in all three alternatives as regards the key length are presented.

In the tables IV, V and VI, the rows C0, C1, C2, C3, and C4 represent values of the established observation categories, and the size of Chi Squared result is flanked by the critical theoretical values.

THE RESULTS OF THE FIRST DIFFUSION SUBSET

Table 4

Plain text -> Cipher text	AES128	AES192	AES256
C0	3261	4868	6493
C1	3068	4611	6252
C2	3714	5561	7332
C3	3125	4623	6092
C4	3216	4913	6599
Lower bound	0.297	0.297	0.297
ChiSquare Result	3.4824	3.8692	4.5209
Higher Bound	13.277	13.277	13.277
Decision	passed	passed	passed

THE RESULTS OF THE SECOND DIFFUSION SUBSET

Table 5

Key -> Cipher text	AES128	AES192	AES256
C0	3267	3229	3269
C1	3064	3050	3015
C2	3638	3688	3614
C3	3112	3146	3248
C4	3303	3271	3238
Lower bound	0.297	0.297	0.297
ChiSquare Result	1.1198	3.0131	4.5209
Higher Bound	13.277	13.277	13.277
Decision	passed	passed	passed

The fact that a block cipher does not pass the diffusion test leads to the conclusion that the initialization stage should be revised.

The results of the dependence tests. They are given in table 6.

DEPENDENCE TESTS RESULTS

Table 6

AES-128				
No. rounds	***NBits***	***TDC***	***TDA***	***TDSA***
1	4.039646	0.062500	0.063119	0.059095
2	16.065028	0.250000	0.251016	0.247624
3	64.251705	1.000000	0.996067	0.991375
4	64.002027	1.000000	0.999305	0.992043
5	63.993756	1.000000	0.999295	0.992012
6	63.995804	1.000000	0.999267	0.991988
7	64.000981	1.000000	0.999284	0.992019
8	63.990952	1.000000	0.999352	0.992030
9	64.003954	1.000000	0.999269	0.992010
10	63.989095	1.000000	0.999342	0.991998

In the case of AES algorithm, the results achieved for the four dependence criteria, after successively performing the rounds, are presented in table VI.

It can be seen that, from the 3rd round on, the algorithm has very strong properties. The results for AES-192, and AES-256 are similar to those illustrated in the table above, for AES-128.

CONCLUSIONS

The cryptographic algorithm represents the core of IT items security. That is why assessing them with reproducible results is essential.

This paper tried to investigate the block algorithm using statistical methods. There have been implemented many types of tests, the purpose of these being the verification of randomness, confusion, and diffusion properties.

Although, the statistical testing represents, maybe, the most important stage in the security assessment of an algorithm, we intend, in the near future, to study the cryptanalysis along with the types of attack.

References

1. **J. Soto, L. Bassham**, 2000, *"Randomness Testing of the Advanced Encryption Standard Finalist Candidates"*, NIST.
2. **A. Rukhin, et. al.**, 2000 , *"A Statistical Test Suite for the Validation of Random and Pseudo Random Number Generators for Cryptographic Applications"*, NIST Special Publication 800-22, .
3. **M. Marin, F. Răstoceanu, R. Dragomir**, 2009 , *"Statistical testing of random number generators"*, A ASES 2009 – Scientific research and education in the air force 20-22 May Brasov, ISBN: 978-973-8415-67-6.
4. **R. Dragomir, M. Marin, F. Rastoceanu, F. Roman**, 2012 *"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.
5. **I. R. Dragomir, D. Filip, M. Marin**, 2012, *"Boolean functions used in cryptology"*, 10th International Symposium on Electronics and Telecommunication, 15-16 November 2012 Timisoara, ISBN:978-1-4673-1174-8, *http://iasmina.cm.upt.ro/work/2012-ISETC/TOT.pdf*.
6. **NESSIE Project** – *New European Schemes for Signature, Integrity and Encryption*, http://cryptonessie.org 2000.
7. **B. Preneel, A. Bosselaers, V. Rijmen, B. Van Rompay, L. Granboulan, J. Stern, S. Murphy, M. Dichtl, P. Serf . Biham, O. Dunkelman, V. Furman F. Koeune, G. Piret, J-J. Quisquater, L. Knudsen, H. Raddum**, 2000, *"Comments by the NESSIE Project on the AES Finalists"*, 24 May 2000.
8. **J. Daemon, V. Rijmen**, 2002, *"The Design of Rijndael"*, Springer 2002, ISBN: 3-540-42580-2.
9. **S. Kavut and M. D. Yucel**, 2001 , *"On Some Cryptogtaphic Properties of Rijndael"*, Springer-Verlag 2001.
10. **M. S. Turan, A. Doganaksoy and C. Calik**, 2006, *"Statistical analysis of synchronous stream ciphers"*, SASC 2006: Stream Ciphers Revisited.
11. **C. E. Shannon**, 1949 , *"Communication Theory of Secrecy Systems"*, Bell Systems Technology Journal, vol. 28, nr. 4, 1949, p. 656-715.
12. **I. R. Dragomir, M. Lazăr**, 2016 *"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