STATISTICAL TESTING OF THE INITIALIZING STAGE OF A BLOCK CIPHER, AS PART OF THE SECURITY ASSESSMENT

Ioana Roxana DRAGOMIR[1], Adriana VLAD[2]

Abstract: Block ciphers are used on a large scale in cryptographic applications. The process of testing the security provided by the block algorithms is very complex and expensive.

The paper focuses on the algorithm which generates the round keys, paying attention to the dependence relation between the master key and the round keys. Approaching some elements belonging to data testing stage, we analyze the two concepts identified by C. E. Shannon, diffusion and confusion, on the mechanism of round key generation.

This way, we evaluate (with statistical precision) the number of round keys and we assess the extent to which a proper statistic report between master key and round keys is reflected in the data processing stage of the cipher.

In order to point out the importance of the applied testing methods and to highlight the findings relevant for the user, we present the experimental study based on AES (Advanced Encryption Standard).

Key words: block ciphers, confusion, diffusion, key schedule, dependence test, frequency test

1. Introduction

Nowadays, the testing of a block cipher has to be carried out in two steps: the initialization procedure which implies round keys generation and the data processing procedure which implies plain text processing (encrypting/decrypting).

Starting from the conclusion that block ciphers have problems with the initialization stage [1], [2] the paper pays special attention to the algorithm for generating the round keys (key scheduling). Our approach is to implement and adjust tests [3], [4] that are considered specific to the data processing stage and to use them in our search for the relation *master key-round keys*.

[1] PhD Student, Faculty of Electronics, Telecommunication and Information Technology, University POLITEHNICA of Bucharest, Romania, e-mail: milita_roxana@yahoo.com
[2] Prof., Faculty of Electronics, Telecommunication and Information Technology, University POLITEHNICA of Bucharest, Romania.
Senior researcher, The Research Institute for Artificial Intelligence, Romanian Academy, Romania, e-mail: avlad@racai.ro, adriana_vlad@yahoo.com.

The initialization procedure - which represents the round keys generation - has an important share as regards the quality of an algorithm, thus the statistical testing of the keys becomes critical. Most attacks rely on finding out the round keys and on exploiting the statistical relations between round keys and master key from which the former ones were generated.

We take into consideration some elements specific to data testing stage (encrypting / decrypting), analyzing the statistic properties, in order to be used in the initialization stage. In this respect, we analyze properties such as confusion and diffusion [5]. In our test procedure master key takes over the role of input data and the round keys stand for output data (as plaintext vs. ciphertext).

To this end, we apply two types of known tests: in order to analyze the diffusion we develop three frequency of occurrence tests and for confusion we implemented dependence tests (looking for degree of completeness property, degree of avalanche and degree of strict avalanche effects).

One of the basic principles that make the block cipher algorithms secure – the confusion – refers to the fact that almost each bit of the ciphertext depends on each bit of the plain text and of the key. As regards the new point of view, namely the relation *master key - round keys*, any minor alteration in the master key is expected to induce, at the same time, a major alteration of the round keys. Otherwise, the cryptanalysts may reach to some conclusions or even sense relations between these elements, all this leading to a significant contraction of key space. In order to assess how this basic criterion is met, several adequate measuring parameters are defined.

A cryptographic function (the algorithm) is considered to be complete when each output bit (in the ciphertext) depends on each input bit (of the plain text) [4]. The avalanche criterion is considered to be met if changing one input bit will change on average half of the output bits. Strict avalanche criterion (SAC), means that changing each input bit implies the change of each output bit with a 50% probability for all possible values of the plain text and of the key.

For the initialization stage (when the round keys are generated) we follow the same requirements; the mechanism of round keys generation is valid if it meets the conditions [6] mentioned below:

- By knowing some relations between two master keys it is “difficult” to infer the relations between the round keys derived from the two master keys;
- By knowing *m* bits from the *r* round key, derived from an unknown master key, it is difficult to find out the other bits of the round keys;
- By knowing all the round keys it is impossible to identify the master key;
- Changing a single input bit from the master key will change half of the bits of each round key – as an average – in an unpredictable manner.

AES block cipher, which is nowadays the encryption standard in force, does not entirely meet these requirements, as shown in our experiments.

The algorithm for generating the keys may become the target of cryptanalytic attacks, such as related-key-attacks [7]. There are ciphers that have weak keys [8], [9]; consequently, the search area diminishes and so does the effort of the cryptanalyst.

The paper is structured as follows. In Section 2, we present a short description of the algorithm which generates the round keys for AES-128, as it is the most frequently used block type symmetrical algorithm. In Section 3, we present the experiments and we describe the testing procedure applied to verify the relationship between the master key and the round keys, by taking into account the same requirements and view-points as in the data processing stage. In Section 4 we display the results we have obtained.

2. Description of the initialization procedure for AES-128 algorithm

In this section, we provide a short description of the algorithm which generates the round keys for AES algorithm. AES is a cipher designed on SPN basis (SPN - substitution permutation network) which supports key lengths of 128 bits, 192 bits and 256 bits. The number of rounds varies according to the key length; thus, for a key of 128 bits there are 10 rounds, for a key having 192 bits there are 12 rounds, and for a key with 256 bits there are 14 rounds.

The criteria taken into consideration by AES authors as regards the algorithm for extending the key [10] are: efficiency and operating memory (*i.e.* the possibility to perform the key extension by using just a small part of the operating memory), performance (*i.e.* high processing speed), removal of symmetries; (to ensure good diffusion) and non-linearity. Diffusion is a transformation meant to dissipate evenly the redundancy, *i.e.* to distribute the redundancy over longer segments of the ciphertext (thus making any statistical testing procedure more difficult).

The round keys [10] are obtained from the encryption key (master key) by a separate procedure which consists of two elements: the key generation and round key selection (see Fig.1).

Each symbol w_i; $i = \overline{0,43}$, from Fig. 1 represents a word that is 32 bits arranged in a column of four bytes. The ensemble [w_0, w_1, w_2, w_3] corresponds to the initial key – the master key; [w_4, w_5, w_6, w_7] corresponds to the first round key etc.

The F function is made up from the two transformations SubWord() function and RotWord() function implemented within the initialization stage:

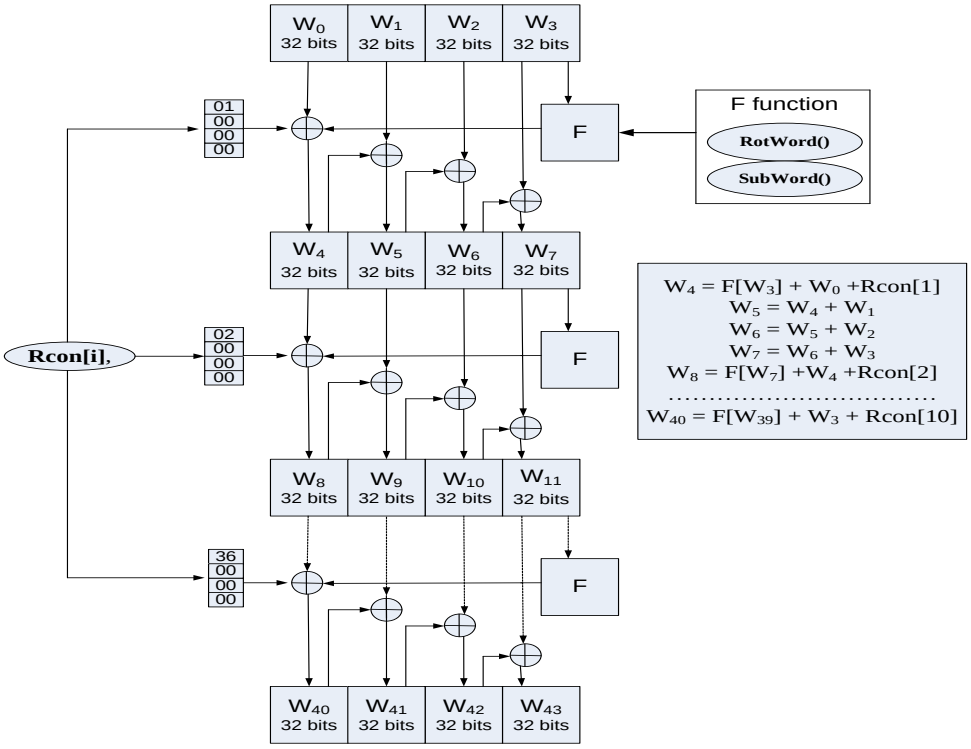

Fig. 1 The structure of the round key generation algorithm

RotWord() function [11] is a cyclic rotation applied to a word of 4 bytes. For example, if we have a word formed of the ensemble $[a_0,a_1,a_2,a_3]$ where a_i $i=\overline{0,3}$ represents a byte, the result after applying function RotWord() is $[a_1,a_2,a_3,a_0]$.

SubWord() function [11] is a function which has as input four bytes. To each byte an S-box is applied; thus, a new byte is obtained.

> *Remarks:* S- box is a non-linear function, (the only non-linear part of the cipher). S-box is the same one used in the algorithm processing stage.

Rcon[ ] is a matrix with 10 columns, each column representing a 32 bits word, respectively the constant Rcon[i] for each round, see Fig.2.

After applying F function, consisting of the two transformations RotWord() and SubWord(), the next step is to add up the result with a round constant value. These round constant values are given in Fig.2, each column (word) in the matrix represents the constant value specific to a round key.

Rcon[i]

01	02	04	08	10	20	40	80	1b	36
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00
R1	R2	R3	R4	R5	R6	R7	R8	R9	R10

Fig. 2 The constant for each round

The first word (32 bits) of each round key is obtained in a different manner from the rest of the words. So, the first 32 bits (the first word) of each round key are obtain as follows (see also Fig.1):

First we applied F function (*i.e.* RotWord and SubWord) to the last word of the master key and then the result is added to the round constant value, thus obtaining the first word in the first round key. Similarly, for the first word of the rest of the round keys, F function is applied to the last word in the previous round key and the result is added with the corresponding round constant value.

It is important to note that the routine which expands the key for the 256 bits cipher is slightly different from the one with 128 bits and 192 bits.

The stages above mentioned were necessary to compute w_4 or the first column from the first round key. The next three columns (w_5, w_6, w_7) will be computed by implementing XOR operation between the column in discussion and the one that had been previously computed, as in Fig.1.

Note that F is a deterministic function composed of a permutation and a substitution, it is a mixing transformation essential in the designing of practical ciphers, with good diffusion and confusion, following in a new way Shannon's suggestions [5], (see also [12], [13] for a presentation of Shannon mixing transforms).

3. Description of experiments

The tests described in this paper will skip over the encryption / decryption function, the main objective being the algorithm which generates the round keys.

Round keys must meet the requirements of good confusion and diffusion properties mentioned in Section 2. The diffusion property is tested by using frequency of occurrence tests and the confusion property is tested by means of dependence tests.

3.1 Frequency of occurrence tests

The purpose of these tests is to establish whether the proportion of 0 values and of 1 values from the string being analyzed is equal to the proportion expected to exist in a binary random sequence compatible to the fair coin model.

The steps in our experiments are:

Step 1: In order to generate the data submitted to test, we took into consideration the following three situations:

Test 1 analyzes the data sets representing round keys with a length of 128 bits obtained from the master key (128 bits) with a low density of 1 symbols. To this end, the following keys are used as master keys:

- One key with all bits 0;
- 128 keys with one bit 1 and the rest of the bits 0 ;
- (128*127)/2 keys with two bits 1 and the rest of the bits 0;

The number of keys submitted to tests is: 1 + 128 + (128*127)/2 = 8257.

Test 2 analyzes the data sets representing round keys with a length of 128 bits obtained from the master key (128 bits) with a very high density of 1 bits.

To this end, the following keys are used as master keys:

- One key with all bits 1;
- 128 keys with one bit 0 and the rest of the bits 1;
- (128*127)/2 keys with two bits 0 and the rest of the bits 1 ;

Test 3 analyzes the data sets representing round keys (with a size of 128 bits) obtained by using a randomly generated master key (with a length of 128 bits).

Step 2: For each of the three tests, a total of 8257 distinct master keys (for AES-128) were used, for each master key the corresponding round keys being generated.

Step 3: The frequency test applied to a round key with 128 bits is considered successfully completed if the $P-value \geq 0.01$, *i.e.* the n occurrence number of the 1 symbol in the round key must be within the range $50 \leq n \leq 78$ (see Table 1). For $n \leq 35$ and $n \geq 93$, the $P-value$ is 0. It can be seen that $P-value \geq 0{,}01$ starting from $n \geq 50$ up to $n \leq 78$; if the occurrence number of bits 1 is in this range of values, the analysed data set passes the test.

Some details regarding the frequency test:

Suppose we have a binary *i.i.d.* (data coming from independently and identically distributed random variables) data set of size N and we want to verify if the data complies with the fair coin model. Let us consider we apply a probability test with H_0 null hypothesis: probability of 1 symbol is $p = 0.5$.

For a binary data set of N bits where the occurrence number of 1 symbol is n, the statistical test value (S_{obs}) is computed as follows:

$$S_{obs} = \frac{n - N/2}{\sqrt{N/4}} = \frac{2n - N}{\sqrt{N}} = \frac{n - n_0}{\sqrt{128}} \tag{1}$$

where n_0 represent the occurrence number of bit 0, ($n + n_0 = N$).

The test values (S_{obs}) are distributed according to the standard Gaussian law (0 mean and 1variance), if H_0 hypothesis is true.

The probability test is applied for a statistical significance level α. If α = 0.01, the test decision is: accept H_0 if $|S_{obs}| \le 2.576$, otherwise reject H_0, that is data are significant for the chosen α statistical level. The constant value 2.576 is the $\alpha/2$ - point value of the standard normal distribution (the 99.5th percentile of the standard normal distribution) [14].

Note that *n* (occurrences for bit 1) is distributed according to the Binomial probability law of $N/2$ mean and $N/4$ variance, if H_0 is true. For *N*=128 the De Moivre-Laplace condition computed as $Np(1-p) = N/4 = 32 >> 1$ can be considered fullfield, so the Binomial distribution can be approximated by the normal law of $N/2$ mean and $N/4$ variance. It results that S_{obs} values from relation (1) are distributed according to the standard normal law. Applying the test decision $|S_{obs}| \le 2.576$ means $n = N/2 \pm 2.576\sqrt{N}/2$ leading to $50 \le n \le 78$.

The corresponding $P - value = erfc\left(\frac{S_{obs}}{\sqrt{2}}\right)$ (2)

where erfc [3] is the complementary error function:

$$erfc(z) = \frac{2}{\sqrt{\pi}} \int_z^{\infty} e^{-u^2} du \quad (3)$$

For each round key generated, the test identifies the *n* number of bits 1. If this number *n* is in the range [50, 78], it means that the test completed with success; otherwise, it means that the test failed.

For example, for a round key with 128 bits where the number of bits 1 is *n* =78, the corresponding probability value is computed as follows:

Compute the statistical test value $S_{obs} = \frac{78-50}{\sqrt{128}} = \frac{28}{\sqrt{128}}$

Compute: $P - value = erfc\left(\frac{S_{obs}}{\sqrt{2}}\right) = erfc\left(\frac{28}{\sqrt{2} * \sqrt{128}}\right) = erfc(1.75) = 0.013328$

where erfc was computed in Wolfram Mathematica.

Table 1

The frequency test for the number of bits '1' and the corresponding P_values

No. of bits '1'	*P – value*	No. of bits '1'	*P – value*	No. of bits '1'	*P – value*
n ≤ 48		**59**	**0.376759**	**70**	**0.288844**
49	0.008010	**60**	**0.479500**	**71**	**0.215925**
50	**0.013328**	**61**	**0.595883**	**72**	**0.157299**
51	**0.021556**	**62**	**0.723674**	**73**	**0.111612**
52	**0.033895**	**63**	**0.859684**	**74**	**0.077100**
53	**0.051830**	**64**	**1.000000**	**75**	**0.051830**
54	**0.077100**	**65**	**0.859684**	**76**	**0.033895**
55	**0.111612**	**66**	**0.723674**	**77**	**0.021556**
56	**0.157299**	**67**	**0.595883**	**78**	**0.013328**
57	**0.215925**	**68**	**0.479500**	79	0.008010
58	**0.288844**	**69**	**0.376759**	n ≥ 80	

Step 4: In order to interpret the test results for test T3 when the master key is a binary *i.i.d.* data set randomly generated, we have analyzed the proportion of binary sequences that passed the statistic test.

The number of binary sequences randomly generated for testing is m = 8257 and the considered significance level is $\alpha = 0.01$. (For example, if from the total of 8257 binary sequences submitted to test, 8040 sequences have, as a results of that test, the *P-value* ≥ 0,01, then the proportion is 8040/8257 = 0.9737.

That means for each round key and each master key we test whether the respective round key passes the test, *i.e. P-value* ≥ 0.01 or not. The test is resumed m = 8257 times, as we considered 8257 randomly generated master keys, and the proportion of the H_0 hypothesis acceptance was recorded for each round. The significance level for each test is $\alpha = 0.01$ meaning that we accept H_0 hypothesis (we decide that the test passes) with a probability $1-\alpha = 0.99$. So, it is expected that 99% of the data sets submitted to the test should pass the test [14].

By probability estimation theory, the range of acceptable proportions is established as below:

$$\beta \pm 3 \cdot \sqrt{\frac{\beta \cdot (1-\beta)}{m}} = 0.99 \pm 3 \cdot \sqrt{\frac{0.99 \cdot (1-0.99)}{8257}}. = 0.99 \pm 0.003284942 \quad (4)$$

where $\beta = 1 - \alpha$, and m is the number of binary sequences submitted to test. If the proportion of test T3 is outside of this range, then it is considered that the analyzed sequences do not pass that test [14].

The confidence range [0.98671506, 0.99328494] was established by using a normal distribution as an approximation of binominal distribution.

For AES-192 and AES-256, the three frequency tests are applied in a similar way; the difference consists in the number of tested keys, namely 18529 and, respectively, 32897, obtained in the following manner:

- For AES-192 we have $1+N+\frac{N(N-1)}{2}=1+192+\frac{192(192-1)}{2}=18529$
- For AES-256 we have: $1+N+\frac{N(N-1)}{2}=1+256+\frac{256(256-1)}{2}=32897$

where N is the size of master key in bits.

3.2 Dependence tests

As we focus on the initialisation stage, testing the dependence criteria means to analyze if the modification of one single bit from the initial key (master key) of the algorithm results in major modifications in the round keys. The criteria necessary to establish dependence are the following: the average number of output bits altered when an input bit is modified; the degree of completeness; the degree of avalanche effect; the degree of strict avalanche effect. These tests were performed according to [4]. The purpose of the dependence test is that when key is smaller, even if the attacker intercepts a large volume of material, he/she should not be able to recover the key because the statistical properties were spread throughout the analyzed material.

Note that the dependence tests described in [4] are specific to data processing stage for a block cipher algorithms and verify the meeting of the criteria of complete transformation, avalanche and strict avalanche over a number of 10000 samples of plaintext, by modifying one bit at a time from the plain text with a constant key (randomly generated). This effect known as avalanche effect, terms used for the first time by Horst Feistel [15], is a concept introduced by Shannon [5].

In order to perform the testing of dependence for the initialisation stage we developed a C++ application according to [4]. To assess the test implementation, we first applied the tests for the data processing stage and we obtained the same results as in [4].

In our experiments the dependence test is carried out in the following steps:

Step1: We shall randomly generate 100 master keys, each having 16 bytes for AES-128, 24 bytes for AES-192 and respectively 32 bytes for AES-256, depending on the variant of chosen algorithm.

First we analyze the AES-128 variant, for each of the 100 keys having 16 bytes each, a bit will be modified at a time, obtaining 128*100=12800 master keys. Each key of the hundred master keys is altered, the changing taking place on “one-by-one” basis.

Step 2: For each round, the round keys obtained from the respective master key are compared to the round keys obtained from the altered master key, the comparison results being saved in two matrices: the dependence matrix and the distance matrix. Based on these matrices, the four criteria are evaluated [4].

Let us assume that we have to compute the dependence A matrix and the distance B matrix of a function $f : \{0,1\}^n \rightarrow \{0,1\}^m$. So a binary data set of *n* size is transformed into a new data set of *m* size. The inputs for f functions are in fact the master keys. For AES-128 $m = n = 128$.

The dependence matrix is a matrix A(n×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.

The distance matrix is a matrix B(n×(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.

The four parameters that were taken into consideration and computed when we verified Shannon's principle for the round key generation algorithm within this test are described in [4] – where the authors apply these tests to data processing stage.

In our experimental study we tried to see if the four criteria considered in the dependence analysis can meet the same requirements considered mandatory in the data processing stage. So, we analysed if the following requirements are accomplished or not when applied to the initialization stage: the average number of output bits altered consequently to a modification of one input is to be larger than 50%; degree of completeness is to be equal with1; degree of avalanche is to be almost equal with 1; degree of strict avalanche is to be almost equal with 1.

4. Experimental results

4.1 The results of the frequency of occurrences tests

The frequency of occurrences tests is implemented according to [3], for a significance level $\alpha = 0.01$. For a chosen α ($\alpha = 0.01$) a certain rate of P-values resulted is expected to show failure (*i.e.* 1% out of the total tested sequences is expected to be rejected). A binary sequence passes a statistic test if the obtained value is P-value $\geq \alpha$.

Table 2 contains the results of the three frequency tests described in Section 3.1, for AES-128, analyzing each algorithm round.

The first table column displays the round number, which in case of AES-128 algorithm may rank from 1 to 10.

$1 + N + \frac{N(N-1)}{2} = 1 + 128 + \frac{128(128-1)}{2} = 8257$ = the total number of round keys;

N is the total number of master key bits.

T1 and T2 show a higher proportion of tests which have the expected behaviour: a good balance between 0 and 1 in the round keys when the master key is "forced" to have low density of 1 symbols for T1 and high density of 1 symbols for T2. Because for T1 and T2 the 8257 master keys are not independent data sets,

we only record the passing ratio. As regards T3, the tested data sets are independent sets and each set is an *i.i.d.* sample (8257 randomly generated master keys), and we can verify if the passing proportion belongs to the interval [0.98671506, 0.99328494] as determined by relation (4). There are two inner columns for T3 which contain the number of sequences that passed the test, considered from the total number 8257, as well as the corresponding final decision (passed or failed).

Table 2

Frequency of occurrence tests results for AES 128

Rounds	T1	T2	T3	
	Proportion	Proportion	Proportion	Decision
1	8040 = 97.37 %	8257 = 100.00 %	8169 = 98.93 %	passed
2	7577 = 91.76 %	8257 = 100.00 %	8166 = 98.90 %	passed
3	8244 = 99.84 %	8230 = 99.67 %	8167 = 98.91 %	passed
4	8192 = 99.21 %	8190 = 99.19 %	8169 = 98.93 %	passed
5	8190 = 99.19 %	8196 = 99.26 %	8163 = 98.86 %	passed
6	8189 = 99.18 %	8211 = 99.44 %	8177 = 99.03 %	passed
7	8189 = 99.18 %	8176 = 99.02 %	8172 = 98.97 %	passed
8	8175 = 99.01 %	8176 = 99.02 %	8181 = 99.08 %	passed
9	8190 = 99.19 %	8170 = 98.95 %	8172 = 98.97 %	passed
10	8163 = 98.86 %	8180 = 99.07 %	8170 = 98.95 %	passed

Tables 3 and 4 present, in a similar manner as Table 2, the frequency of occurrence tests results for AES-192 algorithm and for AES 256, respectively.

Table 3

Frequency of occurrence tests results for AES 192

	T1	T2	T3	
Rounds	Proportion	Proportion	Proportion	Decision
1	18309 = 98.81 %	18529 = 100.00 %	18355 = 99.06 %	passed
2	18499 = 99.84 %	18529 = 100.00 %	18324 = 98.89 %	passed
3	17118 = 92.38 %	18529 = 100.00 %	18323 = 98.89 %	passed
4	18500 = 99.84 %	18497 = 99.83 %	18354 = 99.06 %	passed
5	18431 = 99.47 %	18382 = 99.21 %	18362 = 99.10 %	passed
6	18425 = 99.44 %	18312 = 98.83 %	18337 = 98.96 %	passed
7	18343 = 99.00 %	18294 = 98.73 %	18339 = 98.97 %	passed
8	18376 = 99.17 %	18344 = 99.00 %	18334 = 98.95 %	passed
9	18376 = 99.17 %	18341 = 98.99 %	18334 = 98.95 %	passed
10	18323 = 98.89 %	18392 = 99.26 %	18341 = 98.99 %	passed
11	18331 = 98.93 %	18314 = 98.84 %	18321 = 98.88 %	passed
12	18354 = 99.06 %	18372 = 99.15 %	18354 = 99.06 %	passed

Note that AES-192 has 12 rounds and the number of analyzed sequences is 18529, whereas AES-256 has 14 rounds and the number of analyzed sequences is 32897 (the number of analyzed round keys depends on the key length in bits.

In Table 3 and Table 4, final decision on T3 test means to verify if the passing proportion is within range [0.986891, 0.993109] for AES-192 and range [0.988354262, 0.991645738] for AES-256.

Table 4

Frequency of occurrence tests results for AES 256

	T1	T2	T3	
Rounds	**Proportion**	**Proportion**	**Proportion**	**Decision**
1	32677 = 99.33 %	32897 = 100.00 %	32548 = 98.94 %	passed
2	5380 = 16.35 %	32897 = 100.00 %	32561 = 98.98 %	passed
3	31755 = 96.53 %	32897 = 100.00 %	32591 = 99.07 %	passed
4	31829 = 96.75 %	32897 = 100.00 %	32568 = 99.00 %	passed
5	32794 = 99.69 %	32809 = 99.73 %	32551 = 98.95 %	passed
6	32872 = 99.92 %	32705 = 99.42 %	32571 = 99.01 %	passed
7	32629 = 99.19 %	32570 = 99.01 %	32543 = 98.92 %	passed
8	32634 = 99.20 %	32636 = 99.21 %	32566 = 98.99 %	passed
9	32526 = 98.87 %	32570 = 99.01 %	32592 = 99.07 %	passed
10	32668 = 99.30 %	32624 = 99.17 %	32572 = 99.01 %	passed
11	32653 = 99.26 %	32587 = 99.06 %	32564 = 98.99 %	passed
12	32544 = 98.93 %	32606 = 99.12 %	32562 = 98.98 %	passed
13	32685 = 99.36 %	32692 = 99.38 %	32547 = 98.94 %	passed
14	32589 = 99.06 %	32644 = 99.23 %	32581 = 99.04 %	passed

4.2 The results of dependence tests

We shall first analyze the AES-128 variant; thus, for each key of the hundred keys, made up of 16 bytes each, bits are altered – one at a time - the resulting being 128*100=12800 master keys. For each master key, a round key is generated according to the key expanding algorithm.

Within each round, some indicators are calculated: the average number of output bits which changed when an input bit was altered, the degree of completeness, the degree of avalanche effect, the degree of strict avalanche effect, adapting the procedure described in [4].

For the three variants of AES algorithm, the experimental results obtained for the four dependence criteria after the round key generation are illustrated in Table 5, Table 6 and Table 7.

In Table 5 the first column shows the round number, the second column illustrates – in percentage - the average number of modified bits after altering each master key. For the function to have proper values in respect of the completeness degree, the avalanche effect and the strict avalanche criteria, the requirements in [4] are to be met: the number of modified bits should be over 50%; the value for completeness degree should be 1; the degree of avalanche effect and the degree of strict avalanche criteria must be very close to 1.

Table 5

Dependence test results for AES 128

AES 128				
Rounds	**Number of output bits changed (%)**	**Degree of completeness**	**Degree of avalanche effect**	**Degree of strict avalanche effect**
1	6.522188	0.082031	0.101909	0.056851
2	21.692344	0.324219	0.338943	0.286829
3	33.861719	0.513672	0.529089	0.460332
4	45.831172	0.708984	0.713822	0.646141
5	50.129922	0.781250	0.780861	0.718346
6	50.055078	0.781250	0.779908	0.718397
7	49.943828	0.781250	0.778995	0.718942
8	50.013828	0.781250	0.779135	0.719696
9	49.983359	0.781250	0.778827	0.718368
10	50.061250	0.781250	0.780386	0.718965

We can see in Table 5 that for AES with 128 bits key the number of modified bits reaches the level of 50% in the fifth round, while the other parameters do not exceed the value of 0.78.

Table 6

Dependence test results for AES 192

AES 192				
Rounds	**Number of output bits changed (%)**	**Degree of completeness**	**Degree of avalanche effect**	**Degree of strict avalanche effect**
1	4.348542	0.054688	0.067946	0.037770
2	12.880885	0.183594	0.201264	0.152055
3	20.413229	0.304688	0.318957	0.266340
4	32.023333	0.488281	0.500365	0.439038
5	40.374375	0.615885	0.630850	0.554438
6	45.694323	0.704427	0.712001	0.640448
7	54.090521	0.843750	0.841709	0.776802
8	56.060052	0.875000	0.871710	0.806176
9	55.975937	0.875000	0.870062	0.805895
10	55.942500	0.875000	0.870485	0.805716
11	55.972187	0.875000	0.870916	0.806009
12	56.095417	0.875000	0.871976	0.805068

The same tests applied to data processing stage generate better results [4], [17]; in this case, the number of modified bits reaches the level of 50% in the second round. The other statistical evaluation criteria considered in data processing stage reach the maximum threshold starting with the third round [17].

We can notice in Table 6 that, for AES-192, the number of modified bits exceeds the level of 50% in the seventh round, while the other parameters do not exceed a value of 0.87.

In Table 7, we can remark that the key expanding algorithm for AES-256 achieves a minimum of 50% of modified bits in the nineth round. Although there are 14 rounds, the values for the other three statistic properties are even poorer than those for AES-192.

Table 7

Dependence test results for AES 256

AES 256				
Rounds	**Number of output bits changed (%)**	**Degree of completeness**	**Degree of avalanche effect**	**Degree of strict avalanche effect**
1	3.257344	0.041016	0.050896	0.028523
2	11.370313	0.166016	0.177661	0.142739
3	18.759453	0.287109	0.293116	0.257694
4	24.875937	0.380859	0.388687	0.345579
5	30.910781	0.475586	0.482981	0.430541
6	37.461484	0.577148	0.585336	0.525608
7	43.462617	0.675781	0.677509	0.617668
8	47.974063	0.745117	0.746857	0.682900
9	50.046875	0.781250	0.779170	0.719166
10	50.090742	0.781250	0.780655	0.716966
11	50.066406	0.781250	0.780006	0.719247
12	49.992852	0.781250	0.779479	0.718631
13	50.024297	0.781250	0.779878	0.719464
14	50.066328	0.781250	0.780527	0.718723

For a comparison, Fig.3 presents graphically the evolution of the four criteria for both initialization stage and data processing stage of the AES-256 algorithm. In order to get a unitary image, the graphical representation will be focused on 10 rounds, even if for AES-256 there are more rounds.

The round key generation algorithm is weaker than the data processing algorithm as it can be seen in Fig.3, where the dependence criteria for AES-256 is illustrated. During the course of time, there have been attempts to improve these results by adding various routines to the initializing algorithm. Yet, although the key generation algorithm proved to be weak in terms of statistic test results, due to the manner in which the data processing stage is structured, the algorithm is difficult to break. Note also that any attempt to modify the algorithm by adding a routine may affect the execution speed.

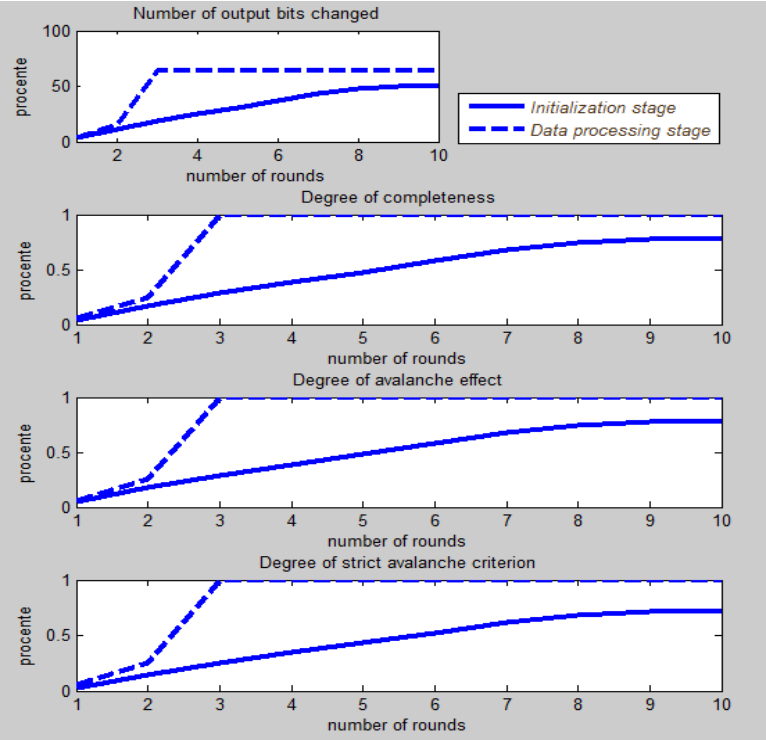

Fig. 3 Graphical representation of the dependence tests results for initialization stage vs data processing stage of AES-256

We can notice that key expanding algorithm reaches the minimum level of 50% for modified bits in the ninth round (see Fig.3). It is worth mentioning that if the same testing approach is applied to the processing stage, the above mentioned level is reached in the third round. The other three criteria of statistic evaluation do not succeed to reach the maximum level earlier than the last round. In the processing stage, these criteria reach the maximum level in the third round [17].

5. Conclusions

The paper presents an investigation of diffusion and confusion concepts as they are revealed through the relationship between *master key- round keys* in a symmetric block cipher. The criteria by which the analysis was carried out in the initialization stage are considered mandatory to be fulfilled (in the specialty literature) for testing the data processing stage of an efficient block cipher. A comparative assessment regarding the diffusion and confusion highlighted in the relation master key - round keys shows poorer results in the initialization stage. The results were illustrated on AES basis. The experimental work induces that the requirements regarding the four dependence criteria (considered necessary in a

security algorithm testing) are a severe desideratum in the initialization stage of an algorithm.

Testing the initialization stage is important for a security assessment of a cipher and these procedures could be useful in the phase of developing an algorithm. By means of these tests one could determine the optimum number of rounds for a block cipher algorithm.

R E F E R E N C E S

[1] *L.May, M. Henricksen, W. Milllan, G. Carter and E. Dawson* Strengthening the Key Schedule of the AES, Springer Verlag Berlin Heidelberg, pp. 226-240, 2002.

[2] *E. Biham* New Types of Cryptanalytic Attacks using related keys, In Advanced Cryptology, Eurocrypt 1993, pages 413-424, Springer-Verlag, 1994.

[3] *A. Rukhin, 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.

[4] *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* Comments by the NESSIE Project on the AES Finalists. 24 May 2000

[5] *C.E. Shannon,* Communication Theory of Secrecy Systems, Bell Systems Technology Journal, vol.28, nr.4, page. 656-715, 1949.

[6] *L. Knudsen* "Practically secure Feistel ciphers", In Fast Software Encryption, Cambridge Security Workshop Proceedings, pages 211-221 Spriger-Verlag, 1994

[7] *J. Kelsey, B. Schneier, D. Wagner,* Key-Schedule Cryptanalysis of IDEA, G-DES, GOST, SAFER and Triple-DES, Advances in Cryptology – CRYPTO 1996, Lecture Notes in Computer Science, vol. 1109, pp 237-251.

[8] *D. W. Davies,* Some regular properties of the DES, Advances in Cryptology – CRYPTO 1992. Plenum Press, pp 89-96.

[9] *L. R. Kndsen,* Cryptanalysis of Loki, Advances in Cryptology – ASIA CRYPT 1991 Springer Verlag, pp 22-35, 1993.

[10] *J. Daemon, V. Rijmen,* The Design of Rijndael, ISBN: 3-540-42580-2 Springer 2002.

[11] *Federal Information Processing Standards Publication 197*, Announcing the Advanced Encryption Standard (AES), 26 November 2001, National Institute of Standards and Technology, U.S. Department of Commerce, November 2001.

[12] *A. Vlad and M. Mitrea*, Cryptographic mixing transformations for image applications, In ROMOPTO 1997: Fifth Conference on Optics International Society for Optics and Photonics, pp. 477-482, 1998.

[13] *A. Vlad and M. Mitrea*, Digital image – protection by means of cryptographic mixing transformations, In ROMOPTO 2000: Sixth Conference on Optics International Society for Optics and Photonics, pp. 560-565, 2001.

[14] *R.E. Walpole and R.H. Myers,* Probability and Statistics for Engineers and Scientists, 4th ed., MacMillan Publishing, NY, 1989.

[15] *H. Feistel*, Cryptography and Computer Privacy, Scientific American pp. 228, 1973.

[16] *D. Toz, A. Doğanaksoy, M. S. Turan*, Statistical Analysis of Block Ciphers, journal: Ulusal Kriptologi Sempozyumu, Ankara, Turkey, pp 56-66, 2005.

[17] *R. Dragomir, M. Marin, F. Rastoceanu şi F. Roman*, Testing block ciphers strength with diffusion method, The 18th International Conference the Knowledge Based Organization, ISSN 1843-6722 page 218-222, 14-16 June 2012.