A Distinguish attack on Rabbit Stream Cipher Based on Multiple Cube Tester

Nasser Ramazani Darmian

University of Aeronautical Science & Technology, Tehran, Iran
nasser.rammazani@gmail.com

Abstract. Rabbit stream cipher is one of the finalists of eSTREAM project which uses 128-bit secret keys. Prior to us, the attacks on Rabbit has been all focused on the bias analysis and the best result showed the distinguishing attack with complexity 2^{136}. Our analysis in this paper, is based on chosen IV analysis on reduced N-S round of Rabbit though using multi cube tester. For this purpose we show for a *mature* cube we could easily identify weak *subcubes* which increase the probability of distinguishing for an unknown secret key. We also represent with 2^{25} complexity, using one iteration of next state function the keystream is completely distinguishable from random.

Keywords: Rabbit Stream Cipher, Cube Attack, Chosen IV Analysis, Distinguishing

1 Introduction

Synchronous stream ciphers are symmetric cryptosystems which are suitable to encrypt message in communications protocols and can be used in hardware or software platforms. Rabbit stream cipher [1] proposed as finalist by ECRYPT, can be served for software application with synchronization purposes. For the first time, Ammuson in [2] showed a distinguish attack with 2^{247} complexity due to a bias in keystream and this complexity was afterwards reduced to 2^{136} by Yi lu [3] and no other considerable result has been obtained. In many protocols, the message is divided into short frames where each frame is encrypted using a different publicly known initialization vector (IV) and the same secret key. The encryption algorithm used must therefore resist against variety of chosen IV attacks. In these attacks, a stream cipher is considered as a black box Boolean function, outputs of which depend on Key and IV variables.

Testing monomials of a Boolean function was first introduced as the name of d-monomial test by Filiol [4]. Saarinen improved Filiols work and modified it using the IV bits instead of the key bits [5]. Following Saarinen, Englund et al worked on d-monomial tests and improved it by introducing three new types of monomial test [6]. AIDA/cube attacks and AIDA/cube testers are two types of recently introduced analyses which are based on Boolean functions and their ANF representation. In 2007, Vielhaber proposed the AIDA (Algebraic IV Differential Attack) and used it to break One.Fivium [7]. Later on, in 2009, the notion of the cube attacks utilized for key recovery, was introduced by Dinur and Shamir in a similar context [8]. At the same year, Aumasson et.al published an extended type of cube attack for distinguishing called cube testers goal of which is to distinguish nonrandomness via property testing.

2 Description of Rabbit

The internal state of the stream cipher consists of 513 bits. 512 bits are divided between eight 32-bit state variables $x_{i,j}$ and eight 32-bit counter variables $c_{i,j}$ where $x_{i,j}$ is the state variable of subsystem j at iteration i, and $c_{i,j}$ denote the corresponding counter variables. There is one counter carry bit, φ_i, which needs to be stored between iterations. This counter carry bit is initialized to zero. The eight state variables and the eight counters are derived from the key at initialization.

The algorithm is initialized by expanding the 128-bit key into both the eight state variables and the eight counters such that there is a one-to-one correspondence between the key and the initial state variables $x_{i,j}$ and the initial counters, $c_{i,j}$. The key $K = (k_{127}, \ldots, k_0)$ is divided into eight subkeys, $K_0 = (k_{15}, \ldots, k_0)\,, K_1 = (k_{31}, \ldots, k_{16})\,, ..., K_7 = (k_{127}, \ldots k_{112})$. The state and counter variables are initialized from the subkeys as follows:

$$x_{j,0} = \begin{cases} K_{(j+1 mod 8)} || K_j & \text{for } j \text{ even} \\ K_{(j+5 mod 8)} || K_{(j+4 mod 8)} & \text{for } j \text{ odd} \end{cases} \tag{1}$$

And

$$c_{j,0} = \begin{cases} K_{(j+4 mod 8)} || K_{(j+5 mod 8)} & \text{for } j \text{ even} \\ K_j || K_{(j+1 mod 8)} & \text{for } j \text{ odd} \end{cases} \tag{2}$$

The system is iterated four times, according to the next-state function defined below, to diminish correlations between bits in the key and bits in the internal state variables. Finally, the counter values are re-initialized according to:

$$c_{j,4} = c_{j,4} \oplus x_{(j+4 mod 8),4} \tag{3}$$

to prevent recovery of the key by inversion of the counter system. The IV setup scheme works by modifying the counter state as function of the IV. This is done by XORing the 64-bit IV on all the 256 bits of the counter state. The system is then iterated four times according to next state function (N-S function) to make all state bits non-linearly dependent on all IV bits. Since the Rabbit in our procedure is used as a black box described by a master polynomial [8], we do not recall next state function. This algorithm generates 128 bit keystream after one iteration of next state function used for encrypting/decrypting plaintext.

3 Algebraic IV Analysis

Let $\mathcal{F}_{\mathrm{n}}$ be set of all function that mapping $\{0,1\}^n \times \{0,1\}^m \to \{0,1\}\,, n, m > 0$ and let $f \in \mathcal{F}_{\mathrm{n}}$. Assume here n is length of key and m is length of IV variables.

The *algebraic normal form* (ANF) of f is the polynomial $f(x)$ over GF(2) in variables $x_1, \ldots, x_n, x_{n+1}, \ldots, x_{n+m}$ (in which $x_1, \ldots, x_n$ show key variables and $x_{n+1}, \ldots, x_{n+m}$ show IV variables) is of the form

$$f(x) = \sum_{i=0}^{2^{n+m}-1} a_i x_1^{i_1} x_2^{i_2} \ldots x_{n+m}^{i_{n+m}} \tag{4}$$

For some $a_0, a_1, \ldots, a_{2^{n+m}-1} \in \{0, 1\}$ and where i_j denotes the j_{th} digit of the binary encoding of i. The term t_I can be indexed by the subset $I \subseteq \{1, \ldots, n+m\}$ which is the monomial consist of all x_i's with $i \in I$, for any function in $\mathcal{F}_n$ ANF of f could be represented algebraically under the form

$$f(x_1, \ldots, x_{n+m}) = t_I.p_{S_I} + q(x_1, \ldots, x_{n+m}) \tag{5}$$

t_I and p_{S_I} are called *cube* and *superpoly* respectively.

3.1 Testing Statistic Properties

Applying statically IV attack, one can choose $I = \{i_1, i_2, \ldots, i_k\}, i_j = 1, ..., m$ and $t_I = x_{i_1} x_{i_2} \ldots x_{i_k}$ and fix other x_i's bits (key and other IV bits) so he has a cube of size k and can obtain values of $f(\ldots, x_{i_1} x_{i_2}, \ldots, x_{i_k}, \ldots)$ for all possible values of $t_I = x_{i_1} x_{i_2} \ldots x_{i_k}$. Now we can define partial ANF in this way

$$f_{K,V}(x) = \sum_{i=0}^{2^k-1} \left(b_i x_{i_1}^{j_1} x_{i_2}^{j_2} \ldots x_{i_k}^{j_k} \right) \tag{6}$$

Where j_l denotes the l_{th} digit of the binary encoding of i and V shows other IV bits not appear in cube set $\{x_{i_1}, x_{i_2}, \ldots x_{i_k}\}$. Coffecients b_is are monomials coefficients and for each selection of key and IV bits we have an independent $f_{K,V}(x)$. Having random properties, each b_i must appear in $f_{K,V}(x)$ with 1/2 probability [9]. It is worth reminding that summing all value of $f_{K,V}(x_{i_1}, x_{i_2}, \ldots, x_{i_k})$ for entire $(x_{i_1}, x_{i_2}, \ldots, x_{i_k}) \in \{0, 1\}^k$, the corresponding superpoly value of p_{S_I} could be constructed. A monomials coefficients vector $(b_0, b_1, \ldots, b_{2^k-1})$ describes an ANF transform of a black box function $f_{K,V}$ in oracle, therefore by analyzing sufficient number of coefficients vectors which are generated with a fixed key and different IVs and exploiting suitable decision rule, we could represent a distinguisher. There are a lot of methods for testing randomness properties in a set of samples but here we use a new powerful test called *multi* χ^2 test [10].

Assume that the set $\{\delta_1, \delta_2, ..., \delta_N\}$ is the sample set of an experiment derived from a population with a balanced binomial distribution. The values $\delta_1, \delta_2, ..., \delta_N$ can be interpreted as the value of an ANF monomial coefficient b_i revealed in $f_{K,V}(x)$ from N different experiments. We expect to have $P(\delta_i = 0) = P(\delta_i = 1) = 1/2$ for $i = 1, \ldots, N$. The output space of the experiment is therefore partitioned into two subspaces $\mathcal{A}_0, \mathcal{A}_1$. In order to test the indistinguishability of δ_i from a random coefficients, the null hypothesis using Pearson's chi-square test assumes $P\left(b_i=0\right)=P\left(b_i=1\right)=1/2$ for any monomial coefficient b_i. The chi-square statistic is then calculated from:

$$\mathcal{X}^2=\frac{\left(O_0-E_0\right)^2}{E_0}+\frac{\left(O_1-E_1\right)^2}{E_1} \tag{7}$$

Where E_i and O_i are the expected and observed frequencies of members of A_i, respectively. By definition, we have $E_0=E_1=1/2$, $O_0+O_1=N$. Therefore, the chi-square statistic becomes:

$$\mathcal{X}^2=\frac{\left(O_1-\frac{1}{2}N\right)^2}{\frac{1}{4}N} \tag{8}$$

Due to the null hypothesis, $\mathcal{X}^2$ is expected to be a random variable with chi-square distribution with one degree of freedom. Corresponding to coefficients vector $\left(b_0,b_1,\ldots,b_{2^k-1}\right)$, the statistic of $\mathcal{X}^2=\sum\limits_{i=0}^{2^k-1}\mathcal{X}_i^2$ has a chi-square with 2^k degree of freedom [10]. This test will be succeded if the value of statistic $\mathcal{X}^2$ become higher than critical value $\mathcal{X}_\alpha^2$. Therefore we could distinguish cipher from random with $1-\alpha$ probability of success in which α is called level of confidence . As we mention above for an unknown key K, we can choose a set $I=\left\{i_1,i_2,\ldots,i_k\right\},i_j=1,...,m$ and $t_I=x_{i_1}x_{i_2}\ldots x_{i_k}$ from IV bits. Therefore for each query of other IV bits we have partial ANF functions $f_{K,V^n}\left(x\right),n=1,\ldots,N$ with corresponding coefficients. Considering sum of all coefficients vectors $\left(B_0,\ldots,B_{2^k-1}\right)=\sum\limits_{n=1}^{N}\left(b_0^n,\ldots,b_{2^k-1}^n\right)$, $\mathcal{X}^2$ statistics will be defined for these coefficients as follow:

$$\mathcal{X}^2=\sum_{i=0}^{2^k-1}\frac{\left(B_i-\frac{N}{2}\right)^2}{\frac{N}{4}} \tag{9}$$

As we have N queries, the complexity of this attack will be $N2^k$ and drastically grew by increasing cube size k.

3.2 Analysis of Biased Monomials Coefficients

In the previous attack we statically used analysis of coefficients vector in a black box stream cipher. It is obvious that sum vector $\left(B_0,\ldots,B_{2^k-1}\right)$ do not have normal distribution and each one has a little bias, also there are always some B_i variables with high and low biases. The maximum and minimum biases in sum vector variables $\left(B_0,\ldots,B_{2^k-1}\right)$ could give us a good criterion to distinguish stream cipher. Now, consider the following lemma could be deduced from [11].

Lemma 1. *Maximum degree coefficient b_{2^k-1} of an $f_{K,V}\left(x\right)$ with k size cube could be computed by XORing outputs of $f_{K,V}\left(x_1,x_2\ldots,x_k\right)$ for all values of* *cube* $(x_1, x_2 \ldots, x_k)$.

We can extend it for each coefficient b_i, $0 \leq i \leq 2^k - 1$. Let b_i be coefficient of monomial in an $f_{K,V}(x_1, x_2 \ldots, x_k)$, therefore one can choose $I' = \{i_1, i_2, \ldots i_l\}$ in which $I' \subseteq I = \{1, 2, \ldots, k\}$ and subcube $t_{I'} = x_{i_1} x_{i_2} \ldots x_{i_l}$ and makes $f_{K,V}(x_{i_1}, x_{i_2}, \ldots, x_{i_l})$. The maximum degree coefficient b_{2^l-1} in $f_{K,V}(x_{i_1}, x_{i_2}, \ldots, x_{i_l})$ could be computed by XORing outputs of $f_{K,V}(x_{i_1}, x_{i_2}, \ldots, x_{i_l})$ for all values of subcube $(x_{i_1}, x_{i_2}, \ldots, x_{i_l})$ which corresponds to b_i in $f_{K,V}(x_1, x_2 \ldots, x_k)$.

Lemma 2. *Let $f_{K,V}(x)$ be a partial ANF with $t_I = x_1 x_2 \ldots x_k$ and $I = \{1, 2, \ldots, k\}$. For each $I' \subseteq I$ in which $t_{I'} | t_I$, all monomials in $f_{K,V'}(x)$ will exactly appear in $f_{K,V}(x)$.*

In other words the partial ANF, $f_{K,V}(x)$ with $I = \{1, 2, \ldots, k\}$ and $t_I = x_1 x_2 \ldots x_k$ include entire monomials of other ANF, $f_{K,V'}(x)$ with $I' \subseteq I$. We call $t_I = x_1 x_2 \ldots x_k$ and the whole possible cubes $t_{I'} | t_I$, mature cube and subcubes respectively.

This is a simple fact that can be deduced from former Lemma 1. Because each b_i coefficient degree in $f_{K,V}(x)$can be interpreted as a maximum degree in an independent different $f_{K,V'}(x)$, all possible cubes $t_{I'} | t_I$ construct the coefficients b_i's in $f_{K,V}(x)$as a result. For example, considering $f(x_1, x_2, x_3) = 1 + x_1 + x_3 + x_2 x_3 + x_1 x_2 x_3$ with mature $t_I = x_1 x_2 x_3$ and subcubes $t_{I_j}, I_j \subseteq I$, we have $t_{I_0} = 0$, $t_{I_1} = x_1$, $t_{I_2} = x_2$, $t_{I_3} = x_3$, $t_{I_4} = x_1 x_2$, $t_{I_5} = x_1 x_3$, $t_{I_6} = x_2 x_3$, $t_{I_7} = x_1 x_2 x_3$. As the maximum degree coefficients for each subcube is $M_0 = 1$, $M_1 = 1$, $M_2 = 0$, $M_3 = 1$,$M_4 = 0$, $M_5 = 0$, $M_6 = 1$, $M_7 = 1$, we can then reperesent a $f(x)$ with maximum degree coefficients obtained from subcubes.

Now, consider a mature cube $t_I = x_1 x_2 \ldots x_k$for a black box stream cipher. Having N different $f_{K,V^n}(x)$ for different IV with corresponding monomials coefficients vectors $\left(b_0^n, \ldots, b_{2^k-1}^n\right)$ computed from a black box stream cipher, we can easily compute sum of these coefficients vectors $(B_0, \ldots, B_{2^k-1}) = \sum_{n=1}^{N} \left(b_0^n, \ldots, b_{2^k-1}^n\right)$.

Basically, If $f_{K,V^n}(x)$ polynomials are random, their coefficients must be appeared with 1/2 probabilities and sum of coefficient variables will have binomial distribution specially with expected value $N/2$ and standard deviation $\sqrt{N}/2$. Any statistic weakness in B_is could be useful for attacker to distinguish cipher. Defining $\epsilon_i = \frac{Bi - \frac{N}{2}}{\frac{N}{2}}, i = 0, \ldots 2^k - 1$ the maximum or minimum biases and its coressponding subcubes could be the best choices of cubes in well-known cube attack. In the other hand by Lemma 2 monomials coefficients $(b_0, \ldots, b_{2^k-1})$ in an ANF $f_{K,V}(x)$ with a mature cube contain entire coefficients of subcubes. As a result the biases of variables of sum vector $(B_0, \ldots, B_{2^k-1})$ denoted by $(\epsilon_0, \ldots, \epsilon_{2^k-1})$ disclose the weak subcubes and we could exploit them to test

whether there is possibility for cube tester to be succeded.
Summing coefficient vectors of N partial ANF $f_{K,V^n}(x), n = 1, \ldots, N$ makes a binomial distribution for every coefficient. Therefore using [12] each monomial coefficient could be distinguished by $N = \frac{\beta}{\epsilon_i^2}$ queries with $P_e = \Phi\left(-\sqrt{\frac{\beta}{2}}\right)$ that β is a small positive integer.
We utilize this technique as preprocessing phase in cube tester in which we find weak subcubes with maximum bias $|\epsilon_{max}|$ for different K and V. If we efficiently try sufficient large set of keys for a mature cube t_I, we will observe that some subcubes get maximum bias multiple times, choosing this subcubes increase the probablity of distinguishing.
As it shown in algorithm1, in online phase, despite of cube tester introduced in [9], we exploit more than one cube to distinguish cipher which means that we extend our variables to impart more nonrandom properties. In this phase a fixed key K is chosen and for mature cube t_I we extract B_is of weak subcubes which were calculated at preprocessing phase. At the end, we exploit multiple $\mathcal{X}^2$ decision rule to distinguish stream cipher.

Algorithm 1 : Online Phase of Cube Tester

```
Input: $K$,$\{i_1, i_2, ...i_v\}$
Output: result of decision
Set $(B_{i_1}, \ldots, B_{i_v})$ to zero
Choose appropriated $N$
for $n \leftarrow 1, N$
    $V^n \leftarrow$ Rand()
    for $(x_1, ..., x_{2^k}) \leftarrow 0, 2^k - 1$
        $(b_0, ...b_{2^k-1}) \leftarrow f_{K,V^n}(x_1, ..., x_k)$
    end for
    $(B_{i_1}, \ldots, B_{i_v}) \leftarrow (B_{i_1}, \ldots, B_{i_v}) + (b_{i_1}, \ldots, b_{i_v})$
end for
$\mathcal{X}^2 = \sum_{j=1}^{v} \frac{\left(B_{i_j} - \frac{N}{2}\right)^2}{\frac{N}{4}}$
If $\mathcal{X}^2 > \mathcal{X}^2_{v,\alpha}$
    Cipher
else
    Random
end If
```

4 Result

We consider this algorithm with different iterarion of next state function in IV-setup step. First, we test a lot of keys and for each key we estimate the maximum biases of monomials cofficients resulted by different use of N-S function. A mature cube by size of $k = 12$ with $N = 6000$ quaries are used and the result presented at table.1.

Table 1. Maximum Biases $|\epsilon_{max}|^2$ for 1,2,3,4 and 5 iterations of next state function

Iteration of N-S function	Maximum Bias
1	$2^{-5.5064} < \lvert\epsilon_{max}\rvert < 2^{-2.6024}$
2	$2^{-5.7179} < \lvert\epsilon_{max}\rvert < 2^{-4.8296}$
3	$2^{-5.7306} < \lvert\epsilon_{max}\rvert < 2^{-4.9809}$
4	$2^{-5.7306} < \lvert\epsilon_{max}\rvert < 2^{-4.9819}$
5	$2^{-5.7306} < \lvert\epsilon_{max}\rvert < 2^{-4.8925}$

Our observation shows that by using one iteration of N-S function the maximum bias will obtain, also distribution function of monomials cofficients is really far from random function and easily distinguishable which corresponds with [12]. In the worst case maximum bias is equal to $|\epsilon_{max}| = 2^{-5.7306}$ which needs $N = 1009$ samples to distinguish it from random with almost $P_e = 13\%$ error probability so there probablly will some subcubes which made cipher distinguishable. Furthermore, according to result on table.1 we could deduce that iterating of N-S function more than twice does not influence distribution of monomials severely.

Applying cube tester, in preprocessing phase for mature cube $t_I = x_1x_2...x_{12}$ we tryed a large set of key K with different V and chose some weak subcubes and specified indexes $(i_1, i_2, ..., i_v)$ of their corresponding B_is in sum vector $(B_{i_1}, B_{i_2}, ..., B_{i_v})$. we also exploited $N = 6000$ queries that result in $6000 \times 2^{12} \approx 2^{25}$ complexity almost.

Finally, in online phase acorrding to algorithm.1 for an unknown fixed key we obtained cofficient sum vector $(B_{i_1}, B_{i_2}, ..., B_{i_v})$ and construct $\mathcal{X}^2$ for these cofficients. we repeat our attack for different fixed unknown keys with $\alpha = 0.05$ and the results are presented in table.2.

Table 2. The distinguishing probabilities for unknown keys after 1,2,3,4 and 5 iteration of next state function

Iteration of N-S function	Probbility of Success
1	90%
2	7% – 10%
3	7% – 10%
4	7% – 10%
5	7% – 10%

Probability of distinguishing shows the probability that an unknown key could be distinguished. As we show in table.2 using one iteration of N-S function make our attack successful with high probability and almost for every unknown key we could easily distinguish keystream from random but using more than one iteration decreases the probability of distinguishing dramatically. Here we reached up to 10% distinguishing probability which means from 10 unknown keys one key is distinguishable. The probability of distinguishing could be increased by testing more keys and IVs in preprocessing steps to find weak subcubes or suitable indexes $(i_1, ..., i_v)$ coresponds with $(B_{i_1}, ..., B_{i_v})$ which have more biases. Also, considering mature cubes with higher length until computation is possible will result higher probability of distinguishing. Furthermore we comprehended that iteration of N-S function more than twice probably will not influence on probability of success.

5 Conclusion

Rabbit is a synchrounous stream cipher proposed by ECRYPT which has been resist against all kind of attacks. In this paper a new kind of cube testers examined on this stream cipher and the results show it might not immune against this types of attacks. This attack was done on reduced iterations of N-S function and the results represent that by using one iteraion of N-S function this algorithm can be distingished from random for any unknown keys. we also introduce mature cube and subcubes concepts and use them to construct new kind of cube testers. Our result is as follow:

- Sum cofficient vector $(B_1, ..., B_k)$ is sum of monomials cofficients of ANF function $f_{K,V^n}(x)$ in which K and V are fixed. Amount of each B_i is equate to maximum degree of it's subcubes and therefore a little bias ϵ_i shows the weakness of the subcube.
- Analysis of Sum cofficient vector for a mature cube of size $t_I = x_1x_2...x_k$ reveals the weak subcubes which could be considered as preprocessing phase. In this step weak subcubes t'_Is and their indexes $\{i_1, ...i_v\}$ for sum vector cofficient $(B_{i_1}, ..., B_{i_v})$ is identified.
- In Online step for a fixed K we test more than one subcubes by well-known X^2 test. The probability of distinguishing can be increased if we test large set of keys and IVs in preprocessing step and using more weak subcubes.

Our test could be extended to multiple weak mature cubes and these will improve the result but the complexity $N2^k$ will increase extremly.

References

1. Boesgaard, M., Vesterager, M., Pedersen, T., Christiansen, J., Scavenius, O.: Rabbit: A new high-performance stream cipher. In: Johansson, T. (ed.) FSE 2003. LNCS, vol. 2887, pp. 307329. Springer, Heidelberg (2003)

2. Aumasson, J.-P.: On a bias of Rabbit (January 2007),
3. Lu, Y., Vaudenay, S., Meier, W., Ding, L., Jiang J.: Synthetic Linear Analysis: Improved Attacks on CubeHash and Rabbit. In: Kim. (eds.) ICISC 2011. LNCS, vol. 7259, pp 248–260. Springer, Heidelberg (2012)
4. Filiol, E.: A New Statistical Testing for Symmetric Ciphers and Hash Functions. In: Deng, R., Bao, F., Zhou, W., Qing S. (eds.) ICICS 2002. LNCS, vol. 2513, pp. 342–353. Springer, Heidelberg (2002)
5. Saarinen, I.: Chosen-IV Statistical Attacks on eSTREAM Stream Ciphers. Report, eSTREAM, ECRYPT Stream Cipher Project (2006)
6. Englund, H., Johansson,T., and Turan, M. S.: A Framework for Chosen IV Statistical Analysis of Stream Ciphers. In: Srinathan, K., Pandu Rangan, C., Yung, M. (eds.) INDOCRYPT 2007. LNCS, vol. 4859, pp. 268–281, Springer, Heidelberg (2007)
7. Vielhaber, M.: Breaking ONE.FIVIUM by AIDA an Algebraic IV Differential Attack. In: Cryptology ePrint Archive, Report 385, (2007)
8. Dinur, I., Shamir, A.: Cube attacks on tweakable black box polynomials. In: Cryptology ePrint Archive, Report 385, (2008). version 20080914:160327
9. Aumasson,J-P., Dinur, I., Meier W. , Shamir, A.: Cube Testers and Key Recovery Attacks On Reduced-Round MD6 and Trivium. In: Dunkelman, O. (eds) FSE 2009. LNCS, vol. 5665, pp. 1–22, Springer, Heidelberg (2009).
10. Vardasbi, A., Salmasizadeh, Mohajeri, J.: On the Multi Chi-square Tests and Their Data Complexity. ISeCure. vol 4. 15–24, (2012)
11. Fischer, S., Khazaei S., Meier, W.: Chosen IV statistical analysis for key recovery attacks on stream ciphers. In: Vaudenay, S.(eds), AFRICACRYPT, vol. 5023, LNCS, pp. 236–245. Springer , Heidelberg (2008).
12. Baignères, T., Junod, P., Vaudenay, S.: How Far Can We Go Beyond Linear Cryptanalysis?. In: Lee, P.J. (eds) ASIACRYPT 2004. LNCS, vol.3329, pp. 432–450, Springer, Heidelberg (2004)