Defence Science Journal, Vol. 62, No. 1, January 2012, pp. 6-10, DOI: 10.14429/dsj.62.1435

Measuring Diffusion in Stream Ciphers using Statistical Testing Methods

[!]Chungath Srinivasan, *Lakshmy K.V., and M. Sethumadhavan

Amrita Vishwa Vidyapeetham, Coimbatore – 641 112, India

[!]E-mail: c_srinivasan@cb.amrita.edu

ABSTRACT

Confusion and diffusion suggested by Claude Shannon are two techniques that symmetric key ciphers should satisfy to prevent cryptanalysis. Diffusion dissipates the statistical properties of the plaintext over the whole ciphertext. For a block cipher, each bit of the output ciphertext block changes with probability one half for any flipped bit in the input plaintext block, implying the cipher to have good diffusion properties. This definition with slight modification can also be applied to stream ciphers but here it is enough to make sure the following: (i) to ensure an overall change in the output keystream with probability half for any flipped bit in the key-IV bit sequence, and (ii) to verify that every bit in the output keystream changes with probability one half for any single bit flip in the key-IV bit sequence. Here we insist on using these tests together for measuring diffusion in stream ciphers. Based on this we have examined the level of diffusion exhibited by some of the eSTREAM candidates and the result is given in this paper.

Keywords: Diffusion, S-Boxes, block ciphers, stream ciphers, keystream, key-IV bit sequence

1. INTRODUCTION

The demand on information security has increased extensively due to the sensitivity of information being exchanged over the public communication channels. Unlike block ciphers, stream ciphers have no standard model for their design and analysis, which leads to cryptographers constructing various models for stream ciphers. From the security perspective, several stream ciphers are found to be vulnerable to cryptanalysis, many of which are statistical analysis[1].

In cryptography, confusion and diffusion are two properties of the operation of a secure cipher which were identified by Claude Shannon[2]. In a block cipher with good diffusion, if one bit of the plaintext is changed, then the ciphertext changes in an unpredictable manner. Cryptographic diffusion test is a kind of statistical test that evaluates a stream cipher or a block cipher for diffusion. The strict avalanche criterion (SAC) builds on the concepts of completeness and avalanche and was introduced by Webster and Tavares[8]. A statistical test for randomness called the SAC test for pseudorandom number generators was presented by Castro[3], *et al.* Turan[4] introduced some structural tests that consider the relation between key-IV and keystream and one of the tests evaluates the diffusion property in stream ciphers.

1.1 Strict Avalanche Criterion of S-boxes

A substitution box (S-box) is an important component used in different cryptographic primitives like stream ciphers, block ciphers, and cryptographic hash functions which provides high nonlinearity if used properly. An m x n S-box[5,8] is a mapping of $f: Z_2^n \to Z_2^m$ that maps an n bit to m bit binary sequence.

The test for checking diffusion in stream ciphers is a mere generalisation of the definition of avalanche criterion on S-boxes[8]. Now we shall brief avalanche criterion on S-boxes and then relate it with diffusion property in stream ciphers. The avalanche criterion on S-boxes can be measured by the following: avalanche effect, strict avalanche criterion-r (SAC-r) and strict avalanche criterion-c (SAC-c), and completeness.

Criterion 1: Avalanche Effect

An n x m S-box, say $f: Z_2^n \to Z_2^m$ exhibits the avalanche effect if and only if

$$\sum_{0\le j\le 2^n-1} W_j = m\,2^{n-1}$$

where $W_j = wt(f(x_j)\oplus f(x_j\oplus e_i^n))$ for each i ('wt' represents the Hamming weight), $x_j \in Z_2^n$ and $e_i^n = (0,0,...,0,1,0,...,0)$ is the 1 bit error vector of n bit length with 1 occupying the i^{th} bit position ($1 \le i \le n$).

This means that whenever a single input bit is complemented then on an average one half of the output bits changes (irrespective of the bit positions).

Criterion 2: Strict Avalanche Criterion-r

An n x m S-box exhibits the strict avalanche criterion-r (SAC-r) if and only if

$$\sum_{0\le j\le 2^n-1} W_j = m\,2^{n-1}$$

where $W_j = wt(f(x_j)\oplus f(x_j\oplus e_i^n)) = \dfrac{m}{2}$ for each j.

This indicates that whenever a single bit in the input to the S-box is complemented then exactly half of the output bits changes. Here, SAC-r implies avalanche effect but the converse need not be true.

*Supported by CSIR Research Fellowship

Received 1 December 2011, online published 23 January 2012

Criterion 3: Strict Avalanche Criterion-c

An *n* x *m* S-box exhibits the strict avalanche criterion-r if and only if

$$\sum_{x \in Z_2^n} f(x) \oplus f(x \oplus e_i^n) = (2^{n-1}, 2^{n-1}, ..., 2^{n-1})$$

for all $i,\ 1 \le i \le n$ (Σ is integer addition).

This indicates that whenever a single bit in the input to the S-box is complemented then every bit in the output changes with probability of one half. Here SAC-c implies avalanche effect and the converse need not. It is to be noted that SAC-r may not imply SAC-c and vice versa.

Criterion 4: Completeness

An *n* x *m* S-box, exhibits the completeness property if and only if

$$\sum_{x \in Z_2^n} f(x) \oplus f(x \oplus e_i^n) = (a_1, a_2, ..., a_m)$$

with $a_i > 0$ for all $i,\ 1 \le i \le m$.

SAC-c implies completeness property but the converse need not be true. Hence we can conclude that it is enough to use SAC-r and SAC-c as the criterion for examining diffusion in a cipher.

1.2 Interpretation of SAC-r and SAC-c using Matrices

For an *n* x *m* S-box, the truth table is a matrix *M* of order $2^n \times m$, where each row is an *m* bit output *f*(*x*) of the S-box. Define weight matrix $M^w$of the same order corresponding to an error vector e_i where each row is $f(x) \oplus f(x \oplus e_i^n)$. Hence for every *n* x *m* S-box there are a total of *n* weight matrices obtained for each error vectors e_i^n ($1 \le i \le n$). These *n* weight matrices are enough to measure SAC-r and SAC-c.

The S-box is to satisfy SAC-r, if the Hamming weight of each row in all the *n* weight matrices equals $m/2$. Unlike SAC-r, if the Hamming weight of each columns for all *n* weight matrices equals 2^{n-1}, then the S-box is to satisfy SAC-c.

2. DIFFUSION PROPERTY IN STREAM CIPHERS

Unlike block ciphers, stream ciphers are said to have good diffusion property if for any single bit flip in the key-IV bit sequence (concatenated bit sequence of key and IV in order) of the cipher corresponding to any fixed key and IV, the keystream changes with probability one half. So, it is enough to verify SAC-r and SAC-c properties for any stream cipher by considering the mapping of the key-IV bit sequence (of size *n*) to the corresponding output keystream (of length *m*). Preferably for better results the value of *m* should atleast *n*, and for simplicity one can choose *m* = *n*. Typically, the size of key-IV bit sequence for stream ciphers is atleast double the size of key, which is 80 bits for hardware oriented stream ciphers[10]. Testing SAC-r and SAC-c for stream ciphers by considering it as an S-box of atleast 160 bits is infeasible, since a weight matrix of order $2^{160} \times m$ (where, m is atleast 160) is to be analysed. But one can test diffusion on stream ciphers by choosing a sample (of size in between 2^{10} and 2^{20}) of inputs from a total of 2^n inputs which are the possible distinct key-IV bit sequences for a stream cipher. For some given input samples, the pseudocodes for measuring the level of diffusion of each bit of the key-IV bit sequence on the keystream of the cipher using SAC-r and SAC-c are given as:

2.1 SAC-r Diffusion Test

Consider the *n* bit length vector e_i = (0,0,…..0,1,0,0….,0), where 1 occupies the *i*th bit position of the vector. Choose a random key-IV bit sequence ($k_1,k_2,…..,k_t,k_{t+1},…..,k_n$), where the first *t* bits represents the key and the remaining bits represents the IV of the stream cipher and then generate an *L* bit length keystream. XOR the error vector e_1 with ($k_1,k_2,…..,k_t,k_{t+1},…..,k_n$) and use the resultant vector as an input to the stream cipher to generate a new *L* bit keystream, which is then XORed with the previously generated *L* bit keystream. Then store the resultant bit stream as a row of a matrix. Repeat this process for *N* different key-IV bit sequences. The matrix obtained is the weight matrix M_1^w of order *N* X *L*. The Hamming weight of each *L* bit length row of the weight matrix is calculated. These *N* values follows binomial distribution *B*(*L*, ½). These values are grouped into five categories as shown in Table 1. The corresponding frequency values follows multinomial distribution with parameters *N* and p_i (*for i*=1,2,...,5), where p_i is the probability of an observed value to lie in the *i*th category. Here the multinomial experiment generalises a binomial experiment by allowing each trial to result in one of the five possible categories. The expected number of trials resulting in category *i* is Np_i. Chi-Square goodness of fit test for binomial distribution with degree of freedom four is applied for analysing the data.

Table 1. Weight matrics

Category	Probability	Expected values
0–523857	0.200224	0.200224×*T*
523858–524158	0.199937	0.199937×*T*
524159–524417	0.199677	0.199677×*T*
524418–524718	0.199937	0.199937×*T*
524719–1048576	0.200224	0.200224×*T*

T is the total number of observed values in an experiment, here $T=2^{16}$

Null hypothesis:

$H_0 : p_1 = p_{10},\ p_2 = p_{20},\ p_3 = p_{30},\ p_4 = p_{40},\ p_5 = p_{50}$

Alternative hypothesis:

H_a: at least one p_i does not equal p_{i0}

The value of p_{i0} depends on the length of the 5 categories defined.

Test statistic value: $\chi^2 = \sum_{i=1}^{5} \frac{(O_i - E_i)^2}{E_i}$

Rejection region: $\chi^2 \ge \chi^2_{\alpha,4}$

$\chi^2_{\alpha,4}$ is the value such that α of the area under the χ^2 curve with *d* (=4) degrees of freedom lies to the right of $\chi^2_{\alpha,4}$. Here α the significance level is chosen to be 0.01.

Repeat the above process for remaining error vectors e_i, *i*=2,…., *n* for the same N key-IV bit sequences and apply chi-square goodness of fit test independently, resulting in *n* number of *p*-values. Preferably all the obtained *n* number of *p*-values should exceed 0.01.

Pseudo code:

```
for $i$=1 to $n$
do
   for $j$=1 to $N$
   do
      Randomly choose key-IV bit sequence as $K_j$ = ($k_1,k_2,\ldots\ldots,k_t,k_{t+1},\ldots..,k_n$)
      Generate $L$ bits of keystream $L_j$ from the cipher using $K_j$
      Construct $K_j^* = K_j \oplus e_i$ and generate $L$ bits of keystream $L_j^*$ from the cipher using $K_j^*$
      $w_j$=Hamming weight $\left(L_j \oplus L_j^*\right)$
      Categorise the value $\boldsymbol{w_j}$
endfor
      Apply chi-square goodness of fit test to the values $w_j$
      Return $p$-value
endfor
```

2.2 SAC-c Diffusion Test

Unlike SAC-r, here we find the Hamming weights of each N bit length column of the weight matrix M_1^w. Then we get L integer values corresponding to L columns of the weight matrix which takes values in between 0 and N (both inclusive). These values should follow binomial distribution $B(L, ½)$. The Chi Square goodness of fit test is applied to estimate the distribution of these L values. Fixing the same N random key-IV bit sequences and repeating the above process using all error vectors e_i, i=2,….,n and applying Chi-square goodness of fit test for each, will result in *n* number of *p*-values. Preferably all *n* number of *p*-values should take value greater than or equal to 0.01.

Pseudo Code:

```
for $i$=1 to $n$
do
      $w_l$=0 ($l$=1 to $L$)
      for $j$=1 to $N$
do
      Randomly choose key-IV bit sequence as $K_j$=($k_1,k_2,\ldots\ldots,k_t, k_{t+1},\ldots..,k_n$),
      Generate $L$ bits of keystream $L_j$ from the cipher using $K_j$
      Construct $K_j^* = K_j \oplus e_i$ and generate $L$ bits of keystream $L_j^*$ using $K_j^*$ as key-IV bit sequence
      $R_j=(L_j \oplus L_j^*)$; $j^{th}$ row of the weight matrix $M=\{M_{jl}\}$ of order $N \times L$
      for $l$=1 to $L$
      $w_l=M_{jl}+w_l$
       endfor
end for
       Categorise the $L$ values $w_l$
       Apply chi-square goodness of fit test to the values $w_l$
       Return $p$-value
end for
```

If the cipher fails the above tests corresponding to any error vector e_i (i=1,2,…,n), then one can retrace the paths affected by those state register(s) occupied by the i^{th} bit of the key-IV bit sequence and modify the cipher accordingly.

3. EMPIRICAL RESULTS

The selection of parameters for the above tests are described here. For both SAC-r and SAC-c diffusion tests, *n* weight matrices each of size 2^{16} x 2^{20} and 2^{20} x 2^{16}, respectively are generated corresponding to *n* error vectors e_i (i=1, 2,…, n). For SAC-r diffusion test the Hamming weights of the 2^{16} rows for each of the *n* matrices are individually grouped into 5 categories (as shown in Table 1) and are evaluated using chi-square goodness of fit test for binomial distribution.

Table 2. Trivium-80 stream cipher: SAC-r/SAC-c results

0.188/0.104	0.068/0.011	0.028/0.071	0.048/0.086	0.000/0.067
0.055/0.012	0.276/0.035	0.046/0.299	0.134/0.079	0.016/0.007
0.006/0.067	0.029/0.026	0.002/0.019	**0.003/0.003**	0.013/0.087
0.010/0.004	0.023/0.130	0.027/0.059	0.008/0.025	0.007/0.179
0.257/0.134	0.171/0.001	0.039/0.017	**0.000/0.000**	0.001/0.015
0.099/0.016	0.011/0.020	0.093/0.087	0.137/0.000	0.109/0.042
0.106/0.025	0.228/0.097	0.065/0.016	0.077/0.153	0.099/0.001
0.005/0.011	0.023/0.015	**0.000/0.000**	0.037/0.046	0.028/0.011
0.039/0.019	0.010/0.057	0.000/0.039	**0.003/0.002**	0.062/0.305
0.033/0.124	**0.004/0.000**	0.093/0.559	0.394/0.024	0.018/0.023
0.022/0.183	0.009/0.149	0.206/0.137	0.326/0.000	0.275/0.152
0.107/0.194	**0.002/0.001**	0.005/0.034	0.077/0.037	0.013/0.011
0.001/0.000	**0.005/0.003**	0.306/0.282	**0.005/0.004**	0.051/0.196
0.238/0.013	0.075/0.206	0.243/0.051	**0.006/0.005**	0.116/0.031
0.026/0.005	0.294/0.028	0.039/0.171	0.037/0.255	0.005/0.016
0.163/0.104	**0.000/0.010**	**0.000/0.007**	0.116/0.252	**0.000/0.000**
0.330/0.012	**0.006/0.003**	**0.002/0.002**	0.163/0.007	0.250/0.274
0.005/0.019	0.211/0.190	0.092/0.128	0.043/0.007	0.008/0.019
0.031/0.029	0.082/0.206	0.148/0.021	0.169/0.050	0.011/0.317
0.009/0.006	0.011/0.028	0.149/0.124	0.084/0.075	0.089/0.052
0.084/0.030	0.230/0.015	0.000/0.010	**0.006/0.006**	0.064/0.040
0.004/0.029	0.144/0.028	**0.003/0.006**	0.012/0.012	0.014/0.000
0.001/0.008	0.342/0.003	0.069/0.231	0.042/0.005	0.270/0.178
0.097/0.031	0.025/0.019	0.039/0.005	0.016/0.240	0.031/0.034
0.002/0.011	0.037/0.073	**0.000/0.005**	0.042/0.019	**0.001/0.002**
0.039/0.034	0.008/0.058	0.007/0.036	0.071/0.159	**0.007/0.000**
0.006/0.005	0.019/0.007	0.098/0.037	0.348/0.082	0.179/0.039
0.050/0.003	0.010/0.188	0.053/0.005	0.165/0.033	0.000/0.041
0.037/0.000	**0.000/0.005**	0.002/0.093	0.001/0.253	0.017/0.003
0.041/0.026	**0.001/0.009**	0.149/0.022	**0.004/0.006**	0.009/0.255
0.022/0.020	0.034/0.008	0.143/0.121	0.003/0.010	0.003/0.018
0.014/0.002	0.014/0.065	0.151/0.010	0.021/0.014	0.159/0.440

Table 3. Grain-80 stream cipher: SAC-r/SAC-c results

0.008/0.012	0.154/0.174	0.042/0.249	0.113/0.037	0.021/0.024
0.045/0.027	0.003/0.030	0.188/0.356	0.074/0.037	0.002/0.097
0.088/0.148	0.109/0.053	0.029/0.005	0.251/0.405	0.040/0.013
0.289/0.100	0.250/0.075	0.081/0.032	0.040/0.048	0.143/0.084
0.131/0.015	0.005/0.053	**0.002/0.001**	**0.005/0.001**	0.131/0.032
0.005/0.044	0.365/0.247	0.008/0.134	0.013/0.016	0.015/0.094
0.038/0.128	0.025/0.003	0.007/0.049	0.010/0.025	**0.003/0.003**
0.157/0.032	0.038/0.112	0.010/0.196	0.019/0.092	0.000/0.228
0.019/0.051	0.099/0.122	0.028/0.035	0.014/0.047	0.000/0.025
0.026/0.025	**0.006/0.001**	0.411/0.120	0.121/0.162	0.036/0.037
0.069/0.121	0.101/0.112	0.023/0.011	0.062/0.241	**0.000/0.002**
0.352/0.054	0.129/0.042	0.098/0.109	0.000/0.087	0.041/0.217
0.051/0.015	0.009/0.020	0.069/0.246	0.084/0.000	0.243/0.060
0.036/0.127	0.052/0.138	0.025/0.015	0.154/0.001	**0.003/0.009**
0.047/0.036	**0.009/0.000**	0.023/0.036	0.181/0.002	0.005/0.074
0.175/0.111	0.030/0.016	0.074/0.127	0.019/0.023	0.292/0.190
0.194/0.245	0.292/0.190	0.019/0.023	0.074/0.127	0.030/0.016
0.175/0.111	0.005/0.074	0.181/0.002	0.024/0.036	**0.009/0.000**
0.047/0.036	**0.003/0.009**	0.154/0.001	0.025/0.015	0.052/0.138
0.036/0.127	0.243/0.060	0.084/0.000	0.069/0.246	0.009/0.020
0.051/0.015	0.041/0.217	0.000/0.087	0.098/0.109	0.129/0.042
0.352/0.054	**0.000/0.002**	0.062/0.241	0.023/0.011	0.101/0.112
0.069/0.121	0.036/0.037	0.121/0.162	0.411/0.120	**0.006/0.001**
0.026/0.025	0.000/0.025	0.014/0.047	0.028/0.035	0.099/0.122
0.019/0.051	0.000/0.228	0.019/0.092	0.010/0.196	0.038/0.112
0.157/0.032	**0.003/0.003**	0.010/0.025	0.007/0.049	0.025/0.003
0.038/0.128	0.015/0.094	0.013/0.016	0.008/0.134	0.365/0.247
0.005/0.044	0.131/0.032	**0.005/0.001**	**0.002/0.001**	0.005/0.053
0.131/0.015	0.143/0.084	0.040/0.048	0.080/0.032	

Similarly for SAC-c diffusion test the Hamming weights of the 2^{16} columns for each of the *n* matrices corresponding to *n* error vectors e_i are individually grouped into 5 categories and are evaluated using chi-square goodness of fit test. The value of *N* should be larger for this test to be applicable. High or low weight values indicate poor diffusion properties of the cipher. Since larger sample size gives accurate result, one may choose $N=2^{16}$, $L=2^{20}$ for SAC-r test and $N=2^{20}$, $L=2^{16}$ for SAC-c test.

We applied the two tests on the eSTREAM candidate ciphers Trivium-80[6] and Grain-80[7]. The bolded i^{th} entry (read row wise) in Table 2 and Table 3 implies that the position(s) (or register(s)) occupied by the i^{th} bit of the key-IV bit sequence in the internal state of the cipher need to be rechecked, as the bit occupying that position in the register does not affect the cipher to cause a change in the keystream. This indicates that these ciphers have poor diffusion for some key/IV bits.

4. CONCLUSIONS

In this paper two new statistical testing methods (SAC-r and SAC-c) are introduced to measure the level of diffusion in the keystream of a stream cipher. A stream cipher should pass both SAC-r and SAC-c diffusion tests for all key/IV bits to confirm proper diffusion. We have noted that stream ciphers Grain-80 and Trivium-80 are not passing these tests for few Key/IV bits. Impact of above observations on the security of the ciphers need to be analysed.

REFERENCES

1. Englund, Håkan; Johansson, Thomas & Turan, Meltem Sönmez. A framework for chosen IV statistical analysis of stream ciphers. *In* Progress in Cryptology– INDOCRYPT 2007 LNCS, 2007, **4859**, pp. 268-81.
2. Shannon, C. Communication theory of secrecy systems. *Bell Syst. Technical J.*, 1949, **28**(4), 656-715.
3. Castro, Julio Cesar Hernandez; Sierra, Jose Maria; Seznec, Andre; Izquierdo, Antonio & Ribagorda, Arturo. The strict avalanche criterion randomness test. *Mathematics and Computers in Simulation*, 2005, **68**(2), 1-7.
4. Turan, M.S.; Doganaksoy, A. & Calik, C. Statistical analysis of synchronous stream ciphers. SASC 2006: Stream Ciphers Revisited 2006.
5. Adams, C & Tavares, S. The structured design of cryptographically good s-boxes. *Journal Cryptology, 1990*, **3**, 27-41.
6. De Cannière, Christophe & Preneel, Bart. Trivium. The eSTREAM Project - eSTREAM Phase 3. http://www.ecrypt.eu.org/stream/triviump3.html [Accessed on April 4, 2011]
7. Hell, M; Johansson, T & Willi, Meier. Grain: A stream cipher for constrained environments. *Int. J. Wireless Mobile Comput.*, 2007, **2**(1), 86-93.
8. Webster, A & Tavares, S. On the design of S-Boxes. *In* Proceedings of the Advances in Cryptology-Crypto85: LNCS, Springer-Verlag, 1986, **219**, pp. 523-34.
9. Liu, Bozhong; Gong, Zheng; Qiu, Weidong & Zheng, Dong. On the security of 4-bit involutive S-boxes for lightweight designs, LNCS, 2011, **6672**, pp. 247-56.
10. ECRYPT: The home page for eSTREAM. The ECRYPT Stream Cipher Project. http://www.ecrypt.eu.org/stream [Accessed on April 4, 2011].

Contributors

Mr Chungath Srinivasan received his MSc (Mathematics) from University of Calicut. Currently, he is working as a faculty in the Amrita Vishwa Vidyapeetham University, Coimbatore.

Ms Lakshmy K.V. received her MSc (Mathematics) from the University of Calicut. She is a full time CSIR funded Research Scholar in Amrita Vishwa Vidyapeetham University, Coimbatore.

Dr M. Sethumadhavan obtained his PhD (Number Theory) from Calicut Regional Engineering College. Currently, he is working as a Professor in the Department of Mathematics and Computer Science, Amrita Vishwa Vidyapeetham University, Coimbatore.