

Detailed Statistical Analysis of Synchronous Stream Ciphers

Meltem Sönmez Turan, Ali Doğanaksoy, Çağdaş Çalık

Institute of Applied Mathematics,
Middle East Technical University, Ankara, Turkey,
{msonmez, aldoks, e110870}@metu.edu.tr

Abstract. The output of synchronous stream ciphers should be indistinguishable from truly random sequences and should not leak any information about the secret key and the internal state of the cipher. In this study, we analyzed the randomness properties of the stream ciphers presented for eSTREAM project. In the first phase of the study, we applied the publicly available NIST test suite to the output sequences. In the second phase, we applied four structural randomness tests to the ciphers. For six of the ciphers, statistical deviations are observed.

Keywords: Synchronous stream ciphers, statistical randomness testing

1 Introduction

The design philosophy of synchronous stream ciphers is inspired by the perfectly secure One Time Pad (OTP) cipher in which the plaintext is encrypted with a random keystream using the XOR operation. OTP is the only cipher known to be unconditionally secure provided that keystream is truly random. Security of OTP lies on the fact that encrypting non-random and statistically weak plaintext with random keystream results in completely random ciphertext.

The requirement of a keystream not shorter than the plaintext, distributing it securely in advance and not recycling the key make OTP impractical. Synchronous stream ciphers, as an approximation to OTP, have the motivation of generating a long pseudo-random keystream using a short random key to overcome these disadvantages. The theoretical security of OTP is no longer available in stream ciphers, but they are very practical and suitable for applications with high throughput requirements. Shortly, the design goal of a synchronous stream cipher is to generate pseudo-random bits which are practically indistinguishable from truly random bits efficiently .

Since the output keystreams are generated using deterministic algorithms, it is sometimes possible to distinguish them from truly random sequences with significant probabilities. In these situations, ciphers are exposed to *Distinguishing Attacks*. In [1], authors argue that weak distinguishing attacks that use long keystream do not represent a security problem in practice. Additionally, there is no commonly accepted keystream length in terms of key size to accept the cipher as broken using the distinguishing attacks. However, existence of polynomial time distinguishers with non-negligible biases violate the design goal of

synchronous stream ciphers. Sometimes, recovering a part of secret key or internal state is possible by improving the ideas of these attacks.

Many different theoretical approaches have been used to find distinguishers to stream ciphers[2][3][4]. Distinguishers are usually found after extensive theoretical analysis. However, statistical randomness testing can be given as another way of finding distinguishers, in an easier way. Usually, In statistical testing, the internal structure of the ciphers is not taken into account. Predefined and cipher independent statistics that are obtained from the keystream are observed, and compared with their theoretical distributions. If a systematic deviation is observed, the output keystream is assumed to be insecure and distinguishable from truly random sequences.

Statistical analysis of stream ciphers are easier and faster compared to theoretical analysis. However, enough importance has not been given to statistical tests. Among eSTREAM candidates, no statistical analysis are reported for the ciphers Achtebahn [5], Decim [6], Dicing [7], Edon80 [8], Lex [9], Mickey [10], Mickey-128 [11], Mir-1 [12], NLS [13], Phelix [14], Polar Bear [15], Pomaranch [16], Salsa20 [17], Sifinks [18], Sosemanuk [19], Trivium [20] and Yamb [21] in algorithm specification documents.

In this study, our main aim is to emphasize the importance of statistical testing, especially in design phase of synchronous stream ciphers. We tested the candidate ciphers for eSTREAM and observed that some of them are very weak statistically. Firstly, we divided our study into two phases. In the first phase, for each cipher keystreams are generated and tested using the publicly available NIST test suite [22]. In the second phase, four structural tests that consider keystream correlations with key, Initialization vector (IV) are applied to the ciphers. Weaknesses are observed for some of the ciphers.

In the next section, a summary of various statistical testing approaches that have been applied by authors of eSTREAM ciphers are given. In Section 3, the keystream testing part is explained in detail. In Section 4, details of structural analysis and description of four tests are presented. Experimental results for both phases are summarized in Section 5. Finally, the conclusion is given in Section 6.

2 Statistical Analysis

In this section, we summarized the statistical testing approaches that have been applied by authors of eSTREAM ciphers.

Anashin et al. [23] proved that the distribution of 32-bit words in ABC output is uniform. The empirical statistical test results given in NIST suite did not indicate any deviation from a random sequence. Also, the authors applied some statistical randomness tests to evaluate the propagation property of the cipher using both Hamming Distance and naive correlation. For a fixed key and for a key varying with each IV pair, 384 such sequences of length 10^6 were obtained and empirically evaluated. Results of NIST Statistical Test Suite and DIEHARD Battery of Tests did not show any deviation from random behavior. As a result of these tests, ABC shows strong propagation properties [23].

In [24], Fubuki and AES have been tested according to their bit diffusion property with small number of rounds. Using 4-round AES and 2-round Fubuki, the diffusion bias is eliminated.

In [25], the keystream of Dragon is tested by the statistical randomness tests given in Crypt-X. Authors applied the frequency, binary derivative, change point, sub-block and runs tests to 30 keystreams of length 8 megabits. Additionally, the sequence and linear complexity tests were applied to 30 streams with 200 kilobits each. Dragon showed no deviation from randomness according to these results. Also, the output of the F-FCSR generator is tested using the NIST Statistical Test Suite [26].

Theoretical validation for diffusion criteria in the initialization state has been done for Grain to defeat statistical chosen-IV attacks [27].

Wu [28] concentrated on distinguishing attacks while analyzing the randomness of HC-256. Keystreams with no linear masking and weakened feedback function are analyzed and it is concluded that distinguishing 2^{128} bits of keystream of HC-256 from a truly random sequence is computationally infeasible.

In [29], it is reported that the output of Hermes8 is tested using the FIBS 140-2 and DIEHARD Battery of Tests and no deviation from randomness is observed.

Vuckovac [30] reported that the output of Mag is tested for patterns in every stage of development by using statistical randomness tests available in ENT, DIEHARD and Crypt-X test suites. According to the results, no deviation from randomness is observed. The cipher Py had also been tested using statistical randomness tests [31]. It is claimed that the output keystream is uncorrelated and statistical tests should not succeed even when more extensive tests are made.

For the cipher Rabbit, the statistical tests from NIST, DIEHARD and ENT suites was applied. The tests were done for both the internal state and the keystream [32]. Also, various statistical tests were applied to the key setup function and also to the reduced version of Rabbit where each state variable has been given in 8 bits. Authors did not find any statistical weakness in any of these cases.

Hong et al. [33] reported that they had applied statistical randomness tests similar to the ones in NIST suite and had not found any weaknesses for TSC-3. Bigeard et al. [34] tested the output of each component of Vest and claimed that individual streams of any of the outputs of Vest accumulators, combined Vest counters and complete Vest ciphers were indistinguishable from truly random sequences.

The randomness property of WG is given in terms of high period, balance, two-level autocorrelation, t-tuple distribution and linear complexity [35]. The keystream generated using ZK-Crypt passed from the statistical randomness tests of NIST and DIEHARD [36].

Another study has been done by Saarinen [37]. He concentrated on the chosen IV attacks to evaluate randomness properties of stream ciphers. Using 10 randomized keys, biases in ciphers Mag, Frogbit, F-FCSR, Decim, Zk-Crypt and Pomaranch have been observed. The ciphers NLS and TSC-3 is reported to fall

into the borderline category. Also, according to [38] the cipher ZK-crypt fails from very simple repetition test.

In the following section, description of keystream and structural testing phases are described.

3 Phase I - Keystream Testing

In the first phase of our analysis, randomness properties of the output keystream is examined. A very natural and general approach in statistical analysis is to generate large amount of keystream and apply certain statistical tests.

The first step in keystream testing is to characterize truly random sequences by a test statistic with known theoretical distribution. Then, the observed and theoretical values are compared to make a conclusion. Many statistical properties may be given to describe the random sequences. Golomb [39] proposed three postulates for the structure of periodic pseudo-random sequences. However, these three postulates are not sufficient to describe random looking sequences, due to their low linear complexities.

A variety of different statistical tests based on different statistics are available in the literature [40] [41] [42]. In this study, we used the NIST [22] Statistical Test Suite which has been used during AES selection [43]. This suite consists of 15 tests namely; frequency, block frequency, runs, longest run, matrix rank, spectral, non-overlapping template matchings, overlapping template matchings, universal test, linear complexity, serial cumulative sums, runs, approximate entropy, random excursions and variants.

It should be noted that these tests are not originally designed to test the security of stream ciphers, rather to evaluate the randomness properties of finite sequences. They do not consider the internal structure, key or IV loading phases of the ciphers.

4 Phase II - Structural Analysis

In the first phase, the randomness properties of the keystream were examined without considering the effects of key and IV bits. In this phase, we tried to measure the correlations between key, IV and a part of keystream. We used four structural tests [44] for analyzing synchronous stream ciphers. The first test, *Key/Keystream Correlation Test*, considers the correlation between key and the corresponding keystream using a fixed IV. Similarly, the second test, *IV/Keystream Correlation Test*, considers the correlation between IV and the corresponding keystream using a fixed key. The third test, *Frame Correlation Test*, considers the correlation between keystreams using different IV values. The last test, *Diffusion Test*, examines the diffusion property of each bit of key and IV. Evaluation in these tests are done using the Chi-Square Goodness of Fit tests.

Let S be a stream cipher with k -bit key K , v -bit initialization vector V , n -bit internal state (s_1, \dots, s_n) and $z_i, (i = 1, \dots)$ be the keystream. Let $S(K, V, l)$ be

the first l bits of the keystream generated by the cipher S initialized by K and V .

In the following sections, the details of the structural tests are presented.

4.1 Key/Keystream Correlation Test

The purpose of this test is to evaluate the bitwise correlation between the key and the first k bits of keystream. High (or low) correlation between key and keystream may enable the cryptanalyst to recover the secret key using the keystream or may reduce the exhaustive search space for key significantly. If the cipher fails from this test, key loading part of the initialization phase should be revised.

In this test, firstly, IV is fixed to a random value to eliminate its effect and m key values are generated randomly. Next, keystream of length k , z_1, \dots, z_k is produced for each key. To evaluate their correlation, key and its corresponding keystream are XORed and weight of the resulting sequence is calculated. The obtained m weight values are grouped into 5 categories depending on the size of key. Then, frequencies of each category are compared to expected values. The pseudo code of the test is given in Algorithm 4.1. For the Chi-square test to be applicable, it is advised to use $m > 100$.

For a secure cipher, distribution of the weights is Binomial with parameters k and $1/2$, i.e. $P(\text{weight} = k) = \binom{n}{k}(1/2)^n$, with success probability $1/2$. Using these probabilities, for $k = 80$, the category limits are chosen as (i) 0 - 35, (ii) 36 - 38, (iii) 39 - 41, (iv) 42 - 44 and (v) 45 - 80. For $k = 128$ bits, the limits are (i) 0 - 58, (ii) 59 - 62, (iii) 63 - 65, (iv) 66 - 69 and (v) 70 - 128.

Small weights indicate that the key and its corresponding keystream are similar, i.e. they are positively correlated. High weights point to a negative correlation between i^{th} bit of key and i^{th} bit of keystream for $i = 1, \dots, k$. However, the test does not consider the correlations between i^{th} bit of key and j^{th} bit of keystream when $i \neq j$.

Algorithm 4.1: KEY/KEYSTREAM CORRELATION(m)

```

Fix  $V$ ;
for  $i \leftarrow 1$  to  $m$ 
  do
    { Randomly choose key  $K$ ;
      Generate first  $k$  bits of keystream,  $M = S(K, V, k)$ ;
       $w_i = \text{weight of } M \oplus K$ ;
    }
  Categorize  $w_i$ 's;
  Apply Chi - Square of Goodness of Fit test;
return ( $p$  - value)

```

4.2 IV/Keystream Correlation Test

The purpose of this test is to evaluate the bitwise correlation between IV and the first v bits of keystream. High (or low) correlation between IV and keystream may lead to generation of keystream without knowing the value of secret key. If the cipher fails from this test, IV loading part of the initialization phase should be revised.

Firstly, the value of key is fixed to a random value and m random IVs are generated. Then, the keystream of length v is produced using each IV value and the fixed key. To evaluate the correlation, IV and its corresponding keystream are XORed and its weight is calculated. Then, m weight values are grouped into 5 categories depending on the size of IV. Then, frequencies of each category are compared to expected values. The pseudo-code of the test is given in Algorithm 4.2.

For a secure cipher, distribution of the weights is Binomial with parameters v and $1/2$. Using these probabilities, the category limits for $v = 64$ are (i) 0 - 28, (ii) 29 - 30, (iii) 31 - 33, (iv) 34 - 35 and (v) 36 - 64. For $v = 80$, (i) 0 - 35, (ii) 36 - 38, (iii) 39 - 41, (iv) 42 - 44 and (v) 45 - 80. For $v = 128$, (i) 0 - 58, (ii) 59 - 62, (iii) 63 - 65, (iv) 66 - 69 and (v) 70 - 128.

Similar to the previous test, small weights indicate that the IV and its corresponding keystream are similar, i.e. they are positively correlated. High weights point to a negative correlation between i^{th} bit of IV and i^{th} bit of keystream for $i = 1, \dots, k$. However, the test does not consider the correlations between i^{th} bit of IV and j^{th} bit of keystream when $i \neq j$.

Algorithm 4.2: IV/KEYSTREAM CORRELATION(m)

```
Fix  $K$ ;  
for  $i \leftarrow 1$  to  $m$   
  do  
  { Randomly choose initialization vector  $V$ ;  
  Generate first  $v$  bits of keystream,  $M = S(K, V, v)$ ;  
   $w_i = \text{weight of } M \oplus IV$ ;  
  Categorize  $w_i$ 's;  
  Apply Chi - Square of Goodness of Fit test;  
  return ( $p$  - value)
```

4.3 Frame Correlation Test

In synchronous stream ciphers, after generating a fixed length keystream called *frame*, IV values are updated. Since IVs are commonly used as counters, two consecutive IV values are similar. The purpose of this test is to analyze the correlation between frames generated with similar IVs. In this test, first a random key and an IV value are chosen, then a keystream of length L is produced.

This procedure is repeated N times with incremented values of IV. Using these keystreams, a matrix of size $N \times L$ is generated and the column weights of the matrix are calculated. The entries of the matrix is evaluated using the Chi-Square Goodness of Fit. The pseudo-code of the test is given in Algorithm 4.3.

Distribution of the weights is approximately normal with mean $N/2$ and variance $N/4$, when N is large. Columns with very high/low weight indicate weaknesses due to insecure resynchronization. If the cipher fails from this test, IV loading part of initialization phase should be revised.

Algorithm 4.3: FRAME CORRELATION(m, l)

Randomly choose K ;
Fix V ;
for $i \leftarrow 1$ **to** m
 do
 { *Generate first l bits of keystream, $M_i = (M_{i_1}, \dots, M_{i_l}) = S(K, V, l)$;*
 { *Increment V ;*
 for $i \leftarrow 1$ **to** l
 do
 { $w_i = \sum_j M_{i_j}$;
 Categorize w_i 's;
 Apply Chi - Square of Goodness of Fit test
 return (p - value)

4.4 Diffusion Test

This test examines diffusion of each bit of key and IV on the keystream. To satisfy the diffusion property, each bit of IV and key should affect the keystream with equal probability. Minor changes in the IV or key should result in random looking changes in the keystream. In the *Diffusion Test*, firstly, random key and IV values are chosen. Using this key and IV, a keystream of length L is generated. Then by changing each bit of key and IV, new keystreams are generated. Then, these keystreams are XORed with the original keystream. Using these vectors, a matrix of size $(k + v) \times L$ is obtained. This procedure is repeated N times and obtained matrices are added in real numbers. The Chi-Square Goodness of Fit test is applied to the entries of the matrix to evaluate diffusion property. The pseudo-code of the test is given in Algorithm 4.4.

For a secure cipher, the entries of the matrix follow a normal distribution with mean $N/2$ and variance $N/4$, when N is large. The category limits using these approximate probabilities are chosen as, (i) 0 - 498, (ii) 499 - 507, (iii) 508 - 516, (iv) 517 - 525 and (v) 526 - 1024, when N is taken as 1024. For this test to be applicable, the value of N should be > 100 .

Entries with high/low value indicate poor diffusion properties of corresponding cells. If the cipher fails from this test, initialization phase of the algorithm should be revised.

Algorithm 4.4: DIFFUSION(N)

Let M be a $(k + v) \times l$ zero matrix;

$$M = \begin{bmatrix} M_1 \\ M_2 \\ \vdots \\ M_{k+v} \end{bmatrix}$$

```
for  $i \leftarrow 1$  to  $N$ 
  do
  { Randomly choose  $K$  and  $V$ ;
     $C = S(K, V, l)$ ;

    for  $j \leftarrow 1$  to  $k$ 
      do
      {  $K' = K \oplus e_j$ ;
         $D = C \oplus S(K', V, l)$ ;
         $M_j = M_j + D$ ;

        for  $j \leftarrow 1$  to  $v$ 
          do
          {  $V' = V \oplus e_j$ ;
             $D = C \oplus S(K, V', l)$ ;
             $M_{k+j} = M_{k+j} + D$ ;

            Categorize entries of  $M$ ;
            Apply Chi – Square of Goodness of Fit test;
            return ( $p$  – value)
```

5 Experimental Results

The statistical analysis both in terms of keystream and structural properties have been done for the synchronous stream ciphers presented for eSTREAM.

5.1 Result of Keystream Tests

Two different evaluation strategies are used in this suite. The first one is related to the proportion of sequences passing a test. This proportion is directly related to the type I error, α , which is usually chosen to be 0.01. Therefore, the expected value of this proportion is 0.99. A lower bound on the proportions can be given as 0.96015. This bound is obtained by a 99% confidence interval for large number of sequences ($n > 100$).

The second strategy is related to the distribution of p-values which should be uniform on between 0 and 1. Uniformity of p-values is tested using the Chi-Square Goodness of Fit tests. The interval between 0 and 1 is divided into 10 equal sub-intervals, expected and observed values are compared and a new p-value is calculated. If this p-value is > 0.00001 then the sequences is considered to be uniformly distributed. For second evaluation to be applicable, at least $50(= 10 \times 5)$ keystreams are needed due to the Chi-square assumption.

For our tests, we generated 100 keystreams of length 2^{20} bits using randomly chosen key and IV pairs. We tested all outputs using the 15 tests with standard parameters. It is also possible to choose test parameters as a function of cipher dependent values, such as word size in word oriented stream ciphers. A total of 100×188 p-values are obtained for each cipher.

The results that indicate weaknesses are summarized in Table 1. Significant deviations are observed for the ciphers Decim, Frogbit and Polar Bear.

<i>Cipher</i>	<i>Test</i>	<i>p-value</i>	<i>proportion</i>
Decim	Block Frequency	0.000000	0.0000
	Runs	0.000000	0.0200
	Longest Run	0.000000	0.9300
	Nonperiodic Templates	0.000000	0.7200
	Overlapping Templates	0.000000	0.7500
	Approximate Entropy	0.000000	0.8900
Frogbit	Frequency	0.000000	1.0000
	Block Frequency	0.000000	0.9200
	Cumulative Sum	0.000000	1.0000
Polar Bear	Frequency	0.000000	0.0000
	Block Frequency	0.000000	0.0000
	Cumulative Sum	0.000000	0.0000
	Runs	0.000000	0.0000
	Longest Runs	0.000000	0.8600
	Nonperiodic Templates	0.000000	0.0000
	Universal	0.000000	0.3600
	Aperiodic Template	0.000000	0.0000
	Serial	0.000000	0.0000

Table 1. The result of NIST tests that indicate weaknesses. There are total of 148 nonperiodic template test results for each cipher, using different templates. Decim fails from 11 of them, whereas Polar Bear fails from all.

5.2 Results of Structural Tests

For each of the cipher, the four structural tests are applied 100 times using different key and IV values. The selection of parameters are described in this section.

For the *Key/Keystream Correlation Test*, $m = 2^{20}$ keys are generated randomly. For each key, keystreams of length k (80 or 128 bits) are generated using the zero vector as IV. Keys and their corresponding keystreams are XORed and their weights are calculated. The weight probabilities are computed using the Binomial distribution. Then, the weights are categorized into 5 groups with approximately equal probabilities and the correlation between key and keystream bits is evaluated using Chi-Square Goodness of Fit tests.

For the *IV/Keystream Correlation Test*, $m = 2^{20}$ IVs and a fixed key are generated randomly. For each IV, keystream of length v (64, 80 or 128 bits) is generated. IVs and their corresponding keystreams are XORed and their weights are calculated. The probability of each weight is computed using the Binomial distribution. The weight values are categorized into 5 groups with approximately equal probabilities and the correlation between IV and keystream bits is evaluated using Chi-Square Goodness of Fit tests.

For the *Frame Correlation Test*, starting with the IV 0x00000001 and incrementing until 0x00100000, 2^{20} keystreams of length 256 bits are generated with a fixed random key. Using these keystreams, a matrix of size $2^{20} \times 256$ is formed and column weights are calculated. The distribution of these weights is approximately normal with mean 2^{19} and variance 2^{18} . Weights are categorized into 5 groups with approximately equal probabilities and evaluated using Chi-Square Goodness of Fit tests.

Finally, for the *Diffusion Test*, a matrix of size $(k+v) \times 256$ is generated using 2^{10} random key and IV pairs. Using the Binomial distribution, the entries of the matrix are categorized into 5 groups with approximately equal probabilities and diffusion of key and IV bits are evaluated using Chi-Square Goodness of Fit tests.

These four tests are applied to the synchronous stream ciphers presented in eSTREAM and the results are given in Table 2. The expected value for each p-value is 0.5. Most of the ciphers support various key and IV sizes. We used key size of 80 and 128 bits for hardware and software oriented stream ciphers, respectively. For further analysis, other key and IV sizes may be considered.

The table shows the p-values obtained from each test. P-values less than 0.01 indicate a possible weakness. Low p-values have been obtained for the ciphers Decim, F-FCRS-8, Frogbit, Mag and Zk-Crypt. For Decim, it is observed that key and the first k bit of keystream are positively correlated. Similar correlation between IV and keystream is also available for the cipher. As the result of *Frame Correlation Test*, deviation from expected distribution is observed. However, the cipher statistically satisfies the diffusion property. For F-FCRS-8, positive correlation between frames is observed. Moreover, lack of diffusion property of IV bits between 66 and 101 causes the cipher to fail from the *Diffusion Test*. According to our results, the cipher Frogbit does not satisfy the necessary diffusion

property and the frames generated using different IVs are correlated. Due to the small IV size of Mag, the *IV/Key Correlation Test* is not applied. For Mag, the desired diffusion property is not satisfied by the IV values. Therefore, it fails from the last two tests. For Zk-Crypt, the 29th and 30th bits of IV and key do not satisfy the desired diffusion.

6 Conclusion

As stated before, the main aim of this study is to emphasize the importance of statistical testing, especially in design phase of synchronous stream ciphers. These predefined and cipher independent tests are easy to implement and running tests takes very short time.

Although analyzing the ciphers only with statistical methods is not enough, it is a part that should not be skipped.

We divided the analysis into two phases. The first phase focuses on the randomness properties of keystream. We used the publicly available NIST test suite with standard parameters. In the second phase, we applied some structural tests that consider the relation between key, IV and keystream.

As result of this study, we obtained some deviations from expected values. From phase I, Decim, Frogbit and Polar Bear fails. It is possible to distinguish the output of Decim using block frequency, runs, longest run, non-periodic templates, overlapping templates and approximate entropy test. The output of Frogbit is distinguished using frequency, block frequency and cumulative sum test. Thirdly, the output of Polar Bear is distinguished using frequency, block frequency, cumulative sum, runs, longest run, nonperiodic templates, universal, aperiodic template and serial test.

From phase II, we obtained positive correlations between key/keystream and IV/keystream for Decim. Also, correlations between keystreams generated using same key and similar IVs are observed for ciphers Decim, F-FCRS-8, Frogbit, Mag and Zk-Crypt. Also, diffusion weaknesses are observed for F-FCRS-8, Frogbit, Mag and Zk-Crypt. Some of these structural weaknesses can be eliminated by applying slight differences in the ciphers.

<i>Cipher</i>	<i>Key Size</i>	<i>IV Size</i>	<i>Key/Keystream Correlation</i>	<i>IV/Keystream Correlation</i>	<i>Frame Correlation</i>	<i>Diffusion</i>
ABC v.2	128	128	0.594847	0.581965	0.499155	0.356969
Achterbahn	80	64	0.447456	0.509330	0.445827	0.462577
CryptMT	128	128	0.467099	0.565633	0.504488	0.340794
Decim	80	64	0.000000	0.000000	0.000000	0.379434
Dicing	128	64	0.510794	0.518198	0.469110	0.279894
Dragon	128	128	0.515234	0.509020	0.493730	0.471513
Edon80	80	64	0.395271	0.490394	0.450119	0.535973
F-FCRS-8	128	128	0.465567	0.497677	0.000000	0.000000
Frogbat	128	128	0.469433	0.605391	0.000000	0.000000
Fubuki	128	128	0.428599	0.660642	0.449259	0.557280
Grain	80	64	0.424768	0.454764	0.498692	0.478812
HC-256	128	64	0.453745	0.546300	0.513012	0.574439
Hermes8	128	128	0.528543	0.504307	0.496917	0.535768
LEX	128	128	0.569770	0.412822	0.507724	0.460434
Mag	128	32	0.534794	-	0.000000	0.000000
Mickey	80	64	0.576635	0.475921	0.608017	0.528977
Mickey-128	128	128	0.525417	0.657516	0.528486	0.517717
Mir-1	128	64	0.518211	0.490121	0.415622	0.658961
NLS	128	128	0.526565	0.472976	0.525124	0.511876
Phelix	128	128	0.506611	0.506333	0.521635	0.469131
Polar Bear	128	128	0.447812	0.482900	0.487471	0.499867
Pomaranich	128	64	0.462254	0.502730	0.591795	0.569247
Py	128	64	0.496108	0.443042	0.522252	0.489706
Rabbit	128	64	0.547722	0.503607	0.608813	0.537224
Salsa20	128	64	0.504370	0.535973	0.510886	0.480442
SFINKS	80	80	0.540354	0.497505	0.529415	0.676244
Sosemanuk	128	64	0.471755	0.507739	0.516827	0.447910
Trivium	80	64	0.479925	0.504355	0.518177	0.533170
TSC-3	128	64	0.521729	0.351436	0.485147	0.637575
Vest-4	128	64	0.526154	0.657904	0.512552	0.427980
WG	128	128	0.532289	0.363903	0.518250	0.527545
Yamb	128	64	0.326160	0.427037	0.497137	0.407011
Zk-Crypt	128	128	0.593001	0.468749	0.000000	0.000000

Table 2. The average of 100 p-values Key/Keystream Correlation, IV/Keystream Correlation, Frame Correlation and Diffusion Correlation Test

References

1. P. Hawkes and G. Rose. On the applicability of distinguishing attacks against stream ciphers. 2002.
2. H. Englund and T. Johansson. A new simple technique to attack filter generators and related ciphers. In *Selected Areas in Cryptography*, pages 39–53, 2004.
3. J. Daemen and G. Van Assche. Distinguishing stream ciphers with convolutional filters. Cryptology ePrint Archive, Report 2005/039, 2005. <http://eprint.iacr.org/>.
4. D. Coppersmith, S. Halevi, and C. S. Jutla. Cryptanalysis of stream ciphers with linear masking. In *CRYPTO*, pages 515–532, 2002.
5. B. Gammel, R. Göttfert, and O. Kniffler. The Achterbahn Stream Cipher. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/001, 2005. <http://www.ecrypt.eu.org/stream>.
6. C. Berbain, O. Billet, A. Canteaut, N. Courtois, B. Debraize, H. Gilbert, L. Goubin, A. Gouget, L. Granboulan, C. Lauradoux, M. Minier, T. Pornin, and H. Sib. Decim, A New Stream Cipher for Hardware Applications. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/001, 2005. <http://www.ecrypt.eu.org/stream>.
7. L. An-Ping. A New Stream Cipher: Dicing. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/001, 2005. <http://www.ecrypt.eu.org/stream>.
8. D. Gligoroski, S. Markovski, L. Kocarev, and M. Gusev. Edon80. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/001, 2005. <http://www.ecrypt.eu.org/stream>.
9. A. Biryukov. A New 128-bit Key Stream Cipher LEX. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/001, 2005. <http://www.ecrypt.eu.org/stream>.
10. S. Babbage and M. Dodd. The Stream Cipher MICKEY (version 1). eSTREAM, ECRYPT Stream Cipher Project, Report 2005/001, 2005. <http://www.ecrypt.eu.org/stream>.
11. S. Babbage and M. Dodd. The Stream Cipher MICKEY-128 (version 1). eSTREAM, ECRYPT Stream Cipher Project, Report 2005/001, 2005. <http://www.ecrypt.eu.org/stream>.
12. A. Maximov. A new stream cipher Mir-1. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/001, 2005. <http://www.ecrypt.eu.org/stream>.
13. G. Rose, P. Hawkes, M. Paddon, and M. W. de Vries. Primitive specification for NLS. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/001, 2005. <http://www.ecrypt.eu.org/stream>.
14. D. Whiting, B. Schneier, S. Lucks, and F. Muller. Phelix, fast encryption and authentication in a single cryptographic primitive. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/001, 2005. <http://www.ecrypt.eu.org/stream>.
15. J. Hastad and M. Naslund. The stream cipher Polar Bear. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/001, 2005. <http://www.ecrypt.eu.org/stream>.
16. C. Jansen and A. Kolosha. Cascade jump controlled sequence generator. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/001, 2005. <http://www.ecrypt.eu.org/stream>.
17. D. J. Bernstein. Salsa20 design. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/001, 2005. <http://www.ecrypt.eu.org/stream>.
18. A. Braeken, J. Lano, N. Mentens, B. Preneel, and I. Verbauwhede. SFINKS: A synchronous stream cipher for restricted hardware environments. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/001, 2005. <http://www.ecrypt.eu.org/stream>.

19. C. Berbain, O. Billet, A. Canteaut, N. Courtois, H. Gilbert, L. Goubin, A. Gouget, L. Granboulan, C. Lauradoux, M. Minier, T. Pornin, and H. Sibert. Sosemanuk, a fast software-oriented stream cipher. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/001, 2005. <http://www.ecrypt.eu.org/stream>.
20. C. De Cannire and B. Preneel. Trivium specifications. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/001, 2005. <http://www.ecrypt.eu.org/stream>.
21. LAN Crypto. Primitive specifications. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/001, 2005. <http://www.ecrypt.eu.org/stream>.
22. A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, and S. Vo. A statistical test suite for random and pseudorandom number generators for cryptographic applications. 2001. <http://www.nist.gov>.
23. V. Anashin, Bogdanov A., Kizhvatov I., and Kumar S. ABC: A New Fast Flexible Stream Cipher. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/001, 2005.
24. M. Matsumoto, H. Mariko, T. Nishimura, and M. Saito. Cryptographic Mersenne Twister and Fubuki Stream/Block Cipher. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/001, 2005. <http://www.ecrypt.eu.org/stream>.
25. E. Dawson, K. Chen, M. Henricksen, W. Millan, L. Simpson, and S. Moon H. Lee. Dragon: A Fast Word Based Stream Cipher. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/001, 2005. <http://www.ecrypt.eu.org/stream>.
26. T. Berger, F. Arnault, and C. Lauradoux. F-FCSR. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/001, 2005. <http://www.ecrypt.eu.org/stream>.
27. M. Hell, T. Johansson, and Willi Meier. Grain - A Stream Cipher for Constrained Environments. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/001, 2005. <http://www.ecrypt.eu.org/stream>.
28. H. Wu. Stream Cipher HC-256. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/001, 2005. <http://www.ecrypt.eu.org/stream>.
29. U. Kaiser. Hermes stream cipher. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/001, 2005. <http://www.ecrypt.eu.org/stream>.
30. R. Vuckovac. MAG My Array Generator (A New Strategy for Random Number Generation). eSTREAM, ECRYPT Stream Cipher Project, Report 2005/001, 2005. <http://www.ecrypt.eu.org/stream>.
31. E. Biham and J. Seberry. Py: A fast secure stream cipher using rolling arrays. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/001, 2005. <http://www.ecrypt.eu.org/stream>.
32. M. Boesgaard, M. Vesterager, T. Christensen, and E. Zenner. The stream cipher rabbit. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/001, 2005. <http://www.ecrypt.eu.org/stream>.
33. J. Hong, D. H. Lee, Y. Yeom, D. Han, and S. Chee. T-function based stream cipher TSC-3. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/001, 2005. <http://www.ecrypt.eu.org/stream>.
34. C. Bigeard, S. O'Neil, B. Gittins, and H. Landman. VEST hardware dedicated stream ciphers. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/001, 2005. <http://www.ecrypt.eu.org/stream>.
35. G. Gong and Y. Nawaz. The WG stream cipher. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/001, 2005. <http://www.ecrypt.eu.org/stream>.
36. C. Gressel, R. Granot, and G. Vago. Zk-crypt - a compact stream cipher and more. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/001, 2005. <http://www.ecrypt.eu.org/stream>.

37. M. O. Saarinen. d-monomial Tests are Effective against Stream Ciphers. 2006.
38. D. J. Bernstein. Does zk-crypt version 1 flunk a repetition test? eSTREAM, ECRYPT Stream Cipher Project, Report 2006/001, 2006. <http://www.ecrypt.eu.org/stream>.
39. S. W. Golomb. *Shift Register Sequences*. Aegean Park Press, Laguna Hills, CA, USA, 1981.
40. D. E. Knuth. *Seminumerical Algorithms*, volume 2 of *The Art of Computer Programming*. Addison-Wesley, 1981.
41. G. Marsaglia. DIEHARD Statistical Tests. <http://stat.fsu.edu/geo/diehard.html>.
42. Information Security Institute. Crypt-X, 1998.
43. J. Soto. Randomness testing of the AES candidate algorithms, 1999.
44. M. S. Turan, A. Doganaksoy, and C. Calik. Statistical analysis of synchronous stream ciphers. *SASC 2006: Stream Ciphers Revisited*, 2006.