Acta Cybernetica **23** (2017) 371–378.

Statistical Analysis of DH1 Cryptosystem

Pál Dömösi[a], József Gáll[b], Géza Horváth[b], and Norbert Tihanyi[c]

In memory of Professor Zoltán Ésik

Abstract

In this paper we shall use some standard statistical methods to test the avalanche effect of a previously introduced cryptosystem based on automata compositions, called DH1 cryptosystem. We have generated sample data of encryption and decryption. In our first set of analysis we simply estimated the probabilities of the atoms of the discrete distribution separately in order to compare them with those of the binomial test distribution. In the second statistical analysis, we turned to a goodness-of-fit test. For this we used the χ^2-test. Thirdly, we assumed that the sample comes from a binomial distribution and we calculated the maximum likelihood estimation of the two parameters. Finally we discuss some well-known further tests on randomness and related results. Our main conclusions based on the statistics all confirm that the avalanche effect is fulfilled.

Keywords: automata network, block cypher, statistics, goodness-of-fit, MLE

1 Introduction

Modern block cyphers are symmetric cryptosystems operating on fixed-length groups of bits, called blocks. These blocks contains at least 128 bits. The cryptosystem transforms the plaintext blocks into cyphertext blocks one by one. In [1] the authors introduced a novel block cypher based on abstract automata and Latin cubes, which is called DH1 cryptosystem in [3]. Another type of cryptosystem based on compositions of automata can be found in [2]. The basic idea of DH1 cryptosystem is to use a giant size finite automaton and a pseudorandom generator. The set of states of the automaton consists of all possible plaintext/cyphertext blocks, and the input set of the automaton contains all possible pseudorandom blocks. The size of the pseudorandom blocks are the same as the size of the plaintext/cyphertext blocks: 128 bits. For each plaintext block the pseudorandom generator generates

[a] Institute of Mathematics and Informatics, College of Nyíregyháza, H-4400 Nyíregyháza, Sóstói út 36, Hungary, E-mail: `domosi@nyf.hu`

[b] Faculty of Informatics, University of Debrecen, H-4028 Debrecen, Kassai út 26, Hungary, E-mail: {`gall.jozsef,horvath.geza`}`@inf.unideb.hu`

[c] Faculty of Informatics, Eötvös Loránd University, H-1117 Budapest, Pázmány Péter sétány 1/C, Hungary, E-mail: `tihanyi.pgp@gmail.com`

DOI: 10.14232/actacyb.23.1.2017.20

the next pseudorandom block, and the automaton transforms the plaintext block into a cyphertext block by the effect of the pseudorandom block. The key is the transformation matrix of the automaton.

$$\mathcal{A} = (\{0, 1, ..., n-1\}, \{0, 1, ..., n-1\}, \delta)$$

δ	0	1	...	$n-1$
0	$c_{0,0}$	$c_{0,1}$	$\cdots$	$c_{0,n-1}$
1	$c_{1,0}$	$c_{1,2}$	$\cdots$	$c_{1,n-1}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$n-1$	$c_{n-1,1}$	$c_{n-1,2}$	$\cdots$	$c_{n-1,n-1}$

- $n = 2^{128}$
- the states $(c_{0,0}, ..., c_{n-1,n-1})$ are numbers between 0 and $n-1$
- each row is a permutation of states (other case decryption is impossible)
- each column is a permutation of states (other case statistical attacks are possible)
- input letters are pseudorandom numbers between 0 and $n-1$
- the key is the transition function itself (+ an initial value: the seed of the pseudorandom number generator (PRNG)).

The following example shows the encryption of a secret message which contains the following 3 blocks: 12993,999833,22212211

- plaintext blocks: 12993,999833,22212211
- suppose the (secret) pseudorandom number blocks are: 2012200, 239993,178
- ciphertext blocks: $c_{2012200,12993}, c_{239993,999833}, c_{178,22212211}$
  $(\delta(2012200, 12993), \delta(239993, 999833), \delta(178, 22212211))$.

The problem with this idea is the following. The size of the transition matrix of the automaton is huge, namely $2^{128} \times 2^{128} \times 16$ bytes, which is impossible to store in the memory or on a hard disk. The solution is to use an automata network. Automata network consists of smaller automata, and it is able to simulate the work of a huge automaton [4]. In [1] the authors introduced a simple automata network which consist of 16 automaton, each of them calculates only one byte of the cyphertext block. Using this simple automata network makes possible simple calculations, but the authors had to introduce an automata network which main rounds contains 4 steps, and each step contains 2 sub steps to have an appropriate avalanche effect. Avalanche effect is an important property for block cyphers, meaning one bit change in the plaintext block should effect significant change in the cyphertext block, and one bit change in the cyphertext block should effect significant change in the corresponding plaintext block. The experimental results shows that 2 main rounds are enough for appropriate avalanche effect, but in this paper we are going to show detailed statistical analysis based on our test data samples.

2 Data and methodology

To test our system, we calculated the number of the identical bytes in two 16 bytes long independent random strings. We have tested 1.000.000 pairs, and saved the result. We also compared 1.000.000 ciphertext block pairs, where the corresponding plaintext blocks had just 1 bit difference. Finally we compared 1.000.000 plaintext block pairs, where the corresponding ciphertext blocks had just 1 bit difference.

Table 1 : Frequency table of the four samples

identical bytes	EN1R	DE1R	EN2R	DE2R
0	915924	916422	938843	939081
1	43064	42710	59403	59145
2	22670	22397	1717	1746
3	880	921	37	28
4	11050	11064	0	0
5	410	396	0	0
6	179	225	0	0
7	11	4	0	0
8	5574	5594	0	0
9	125	136	0	0
10	72	89	0	0
11	3	1	0	0
12	36	40	0	0
13	0	0	0	0
14	1	1	0	0
15	0	0	0	0
16	0	0	0	0

Basically we have generated this way 4 different samples: on the one hand we had samples obtained after encyption (encoding, denoted by EN) and decryption (decoding, denoted by DE), on the other hand after 1 round and 2 rounds (denoted by 1R and 2R) of encryption or decryption. Hence we shall refer to the 4 samples as EN1R, EN2R, DE1R, DE2R, respectively. (See Table 1, where we show the frequencies of the possible values –i.e. the number of identical bytes– of the distributions for all samples.)

Based on the generated samples we considered three different statistical questions to analyse the distribution of the number of different blocks in the pairs. But the main aim behind all questions, of course, was to check wether the avalanche effect can be confirmed in our case. Clearly, in an ideal situation – i.e. where we have an appropriate avalanche effect – one should get a binomial distribution with parameters $n = 16$ and $p = 1 - 1/256$ for the generated data, since in that case one can get no additional information from the data about the coding method. Therefore in what follows we shall call the binomial distribution with the above parameters simply the 'reference distribution'. With different ways we analyze whether our data show significant difference from the reference distribution or not.

For what follows we shall denote the probabilities of the test (reference) distribution by $p_i^{(0)}$, for $i = 0, 1, \ldots, 16$, whereas the probabilities of the real (true) distribution will be denoted by p_i^*, for $i = 0, 1, \ldots, 16$.

In our first set of analysis we simply estimated the probabilities of the 16 atoms of the discrete distribution separately. We calculated the point estimate of the probabilities, furtherore, considering the interval estimate of the probabilities we used confidence level $\alpha = 0,999$, i.e. 99,9% and calculated the maximum value of the margin of errors, which has the form

$$\Delta = \frac{1}{2} z_{1-\frac{\alpha}{2}} \sqrt{\frac{1}{n}},$$

where $z_{1-\frac{\alpha}{2}}$ is a quantile of the standard normal distribution of order $1 - \frac{\alpha}{2}$ and n is the sample size. It is well known that the margin of error takes its maximum for probability 1/2. Since we have a large sample size we decided to fix the confidence level at a very high value, so that small differences are indicated. This way one can see the difference between the probabilities obtained from the reference binomial distribution and the estimated probabilities from the sample.

In the second statistical analysis, we turned to a goodness-of-fit test. For this we used the χ^2-test of goodness of fit with the well-known test statistics

$$\chi^2 = \sum_{i=1}^{k} \frac{(f_i - f_i^*)^2}{f_i^*},$$

where f_i and f_i^* are the observed and the expected frequencies (the latter one being based on the test distribution) for category (probability) i, $i = 1, \ldots, k$, respectively. The aim is of course to check if the null hypothesis can be confirmed according to what the theoretical distribution of the population what the data is coming from coinsides the reference binomial distribution, which is our test distribution.

In our third analysis, we assumed that the sample comes from a binomial distribution (based on the results obtained to the previous questions) and we calculated simply the maximum likelihood estimation of the two parameters of the distribution and compared them to those of the reference distribution.

3 Statistical results

Before turning to the analysis it is worth to mention that one should check the basic properties of the sample. In other words, since we will use some standard statistical methods which are all based on independent and identically distributed sample (i.i.d. sample) one should verify these properties. Which concerns our case, either after one round or two rounds and either in case of encryption or decryption one can clearly see that the generation method of the data assures us that on the one hand the data element show no independence on the other hand their (theoretical) distribution is the same.

Table 2 : Difference of the point estimates and the theoretical values (i.e. $\hat{p}_i - p_i^{(0)}$, $i = 0, 1, \ldots, 16$)

	EN1R	DE1R	EN2R	DE2R
0	**-2.337318e-02**	**-2.287610e-02**	-4.551571e-04	-2.170959e-04
1	**-1.587231e-02**	**-1.622635e-02**	4.667083e-04	2.086489e-04
2	**2.093660e-02**	**2.066358e-02**	-1.642037e-05	1.257791e-05
3	8.482781e-04	8.892772e-04	5.277280e-06	-3.722757e-06
4	**1.104961e-02**	**1.106360e-02**	-4.043097e-07	-4.043097e-07
5	4.099966e-04	3.959962e-04	-3.805267e-09	-3.805267e-09
6	1.790002e-04	2.250000e-04	-2.735813e-11	-2.735813e-11
7	1.100001e-05	4.000000e-06	-1.532668e-13	-1.532668e-13
8	5.574006e-03	5.594000e-03	-6.761772e-16	-6.761772e-16
9	1.250001e-04	1.360000e-04	-2.357045e-18	-2.357045e-18
10	7.200007e-05	8.900000e-05	-6.470319e-21	-6.470319e-21
11	3.000003e-06	1.000000e-06	-1.384025e-23	-1.384025e-23
12	3.600004e-05	4.000000e-05	-2.261480e-26	-2.261480e-26
13	-2.728784e-29	-2.728784e-29	-2.728784e-29	-2.728784e-29
14	1.000001e-06	1.000000e-06	-2.293096e-32	-2.293096e-32
15	-1.199004e-35	-1.199004e-35	-1.199004e-35	-1.199004e-35
16	-2.938736e-39	-2.938736e-39	-2.938736e-39	-2.938736e-39

Which concerns the point estimations for the four samples, Table 2 contains the results, namely: we show the difference of the point estimates and the theoretical values (obtained from the test distribution). In other words we show $\hat{p}_i - p_i^{(0)}$ for all $i = 0, 1, \ldots, 16$, where $\hat{p}_i$ is clearly the point estimate (namely the relative frequency) of p_i^*. One can compare it with the maximum value of the margin of error Δ descibed in the previous section. With $\alpha = 0,999$ we obtain $\Delta = 0,001545116$. We can see from the table that after 1 round some of the estimates have a relatively large difference from the theoretical value, namely larger than 10^{-2} and hence larger than Δ both in case of encryption or decryption. However, after 2 rounds the results are much better since the largest difference at issue is still clearly under 10^{-3}. Thus we can conclude that after two rounds the generated data do not show difference from the theoretical test distribution, with 99,9% of confidence one could not differentiate between the test probabilities and the obtained empirical probabilities. Thus after four rounds the cryptosystem in this way show to fulfill the appropriate avalanche effect.

The results obtained from the χ^2-test can be seen in Table 3. Note that due to the large sample size we had the following concern. One cannot generally hope a clear confirmation of the null hypothesis, since very small differences of the distributions may lead to the rejection of the null hypothesis in such a case. (That is why sometimes the P-values are used only as an indicator: choosing different test distributions the one giving the largest P-value is accepted even if it does not show perfect fit by the test.) However, the results gave a clear picture, namely they lead to the same conclusions as in the previous analysis: after 2 rounds with both samples we cannot reject the null hypothesis that the data comes from the reference distribution. This again confirm that the cryptosystem seems to fulfill the avalanche effect.

Table 3 : Results obtained from the χ^2-test

	EN1R	DE1R	EN2R	DE2R
test stat.	**4.366657e+19**	**4.367992e+19**	5.357948e-06	1.725129e-06
P-values	**≈0 (<10e-10)**	**≈0 (<10e-10)**	≈ 1	≈ 1
conclusion	$\mathbf{H}_1$	$\mathbf{H}_1$	H_0	H_0

Finally, Table 4 shows the results obtained by the maximum likelihood estimations of the two parameters of the binomial distribution (assuming that the data is from the family of binomial distributions). For the test distribution we have $p = 1 - 1/256 \approx 0,9960938$ and $N = 16$. The results after two rounds support again the acceptence of the reference distribution as the real one, since the errors in the estimates are less than 10^{-4}, which is quite satisfactory.

Table 4 : Results obtained by the maximum likelihood estimations

	EN1R	DE1R	EN2R	DE2R
p	**0.9569296**	**0.9566545**	0.9960694	0.9960817
N	**16.52644**	**16.53131**	15.99994	15.99997

3.1 The Lempel-Ziv, Sárközy and Mauduit randomness tests

One of the criteria used to evaluate the AES candidate algorithms was their demonstrated suitability as random number generators. That is, the evaluation of their output utilizing statistical tests should not provide any means by which to distinguish them computationally from a truly random source. In order to test our cryptosystem we performed some basic randomness testing such as the Lempel-Ziv test and Sárközy and Mauduit methods. Data compression methods are very good starting point for testing pseudo randomness of a finite binary string. Applying the Lempel-Ziv test we were not able to distinguish the output of our cryptosystem from true random sources. In order to fulfill further requirements we performed the Sárközy and Mauduit methods [5, 6] so that to study the behaviour of pseudorandom sequences generated by our cryptosystem. Let $E_N = \{e_1, e_2, \ldots, e_N\} \in \{-1, +1\}^N$ represent a finite binary sequence. Let us define

$$U(E_N, M, a, b) = \sum_{j=1}^{M} e_{a+jb}.$$

The *well-distribution measure* of E_N is defined by

$$W(E_N) = \max_{a,b,t} |U(E_N, t, a, b)| = \max_{a,b,t} \left| \sum_{j=1}^{t} e_{a+jb} \right|$$

where the maximum is taken over all a, b, t such that $a \in \mathbb{Z}, b, t \in \mathbb{N}$ and $1 \leq a+b \leq a+tb \leq N$. Furthermore let us define

$$V(E_N, M, D) = \sum_{n=0}^{M-1} e_{n+d_1}\ e_{n+d_2} \dots e_{n+d_k}.$$

The *correlation measure of order k* of E_N is defined by

$$C_k(E_N) = \max_{M,D} |V(E_N, M, D)| = \max_{M,D} \left| \sum_{n=0}^{M-1} e_{n+d_1}\ e_{n+d_2} \dots e_{n+d_k} \right|$$

where the maximum is taken over all M and $D = (d_1, \dots, d_k)$ such that $0 \leq d_1 \leq \dots \leq d_k \leq N - M$. The goodness of a PRNG is determined by the order of $W(E_N)$ and $C_k(E_N)$. Our first results on the issue showed that we were not able to distinguish the output of our cryptosystem from true random sources by analyzing the deviation of $W(E_N)$ and $C_k(E_N)$.

There are many different statistical methods for testing the pseudorandomness of a binary string. For instance, The National Institute of Standards and Technology (NIST) published a statistical package consisting of 15 statistical tests that were developed to test the randomness of arbitrarily long binary sequences produced by either hardware or software based cryptographic random or pseudorandom number generators. Our latest (positive) test results confirm that it is meaningful and hopeful to run further tests on the cryptosystem in this direction. Note that according to the first few test results the DH1 cryptosystem successfully passed the criteria of NIST test so we would like to continue our research in this direction.

4 Conclusions

The results from the statistical estimations and tests show that the distributions of the 3 samples are the same with the same parameters, their distribution coincides with the theoretical binomial distribution, which means that the cryptosystem has an appropriate (efficient, statistically significant) avalanche effect. The first few statistical test results suggest that the output of the cryptosystem can not be distinguished from true random sources by statistical tests.

References

[1] P. Dömösi, G. Horváth: *A novel cryptosystem based on abstract automata and Latin cubes*, Studia Scientiarum Mathematicarum Hungarica, 52(2)(2015):221–232.

[2] P. Dömösi, G. Horváth: *A novel cryptosystem based on Gluškov product of automata*, Acta Cybernetica, 22(2015):359–371.

[3] P. Dömösi, J. Gáll, G. Horváth, N. Tihanyi: *Some remarks on the DH1 Cryptosystem based on automata compositions*, in preparation.

[4] P. Dömösi, C. L. Nehaniv: *Algebraic theory of automata networks: An introduction*, ser. SIAM monographs on Discrete Mathematics and Applications, vol. 11, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2005, doi 10.1137/1.9780898718492.

[5] C. Mauduit, A. Sárközy: *On finite pseudorandom binary sequences I : Measure of pseudorandomness, the Legendre symbol*, Acta Arithmetica, 82(1997), 365-377.

[6] C. Mauduit, A. Sárközy: *On finite pseudorandom binary sequences II : The Champernowne, Rubin-Saphiro, and Thue-Morse sequences, a further construction*, J. Number Theory 73(2) (1998), 256-276.