From: "mrobshaw" <mrobshaw@supanet.com>
To: <AESround2@nist.gov>
Cc: "Jan-Ove Larsson" <JLarsson@rsasecurity.com>,
"Jakob Jonsson" <JJonsson@rsasecurity.com>
Subject: statistical testing of RC6
Date: Fri, 7 Apr 2000 18:45:52 +0100
X-MSMail-Priority: Normal
X-Mailer: Microsoft Outlook Express 5.00.2314.1300
X-MimeOLE: Produced By Microsoft MimeOLE V5.00.2314.1300

Dear Sirs,

Please find attached a summary of some additional statistical testing that we performed on RC6. This was in response to an earlier NIST report on the statistical testing of the AES candidates.

In that first report, a few tests were mentioned for which the behavior of RC6 was briefly questioned. NIST later concluded that the results for RC6 had been reasonable. The experiments described in this note provide further support for this conclusion.

Best wishes,
Matt Robshaw

On the Statistical Testing of RC6

Jakob Jonsson[1], Jan-Ove Larsson[1], and Matthew J.B. Robshaw[2]

[1] RSA Security, Arenavägen 29, SE-121 29 Stockholm
{jjonsson,jlarsson}@rsasecurity.com
[2] 88 Hadyn Park Road, London, W12 9AG
mrobshaw@supanet.com

Abstract. Following publication of NIST results on the statistical testing of the AES first round candidates, we provide the results of our own statistical testing on RC6.

1 Introduction

NIST has published a document [3] called "Randomness testing of the AES Candidate Algorithms". The aim of this statistical testing is to verify the suitability of the different AES candidates as the basis of a pseudo-random number generator. Of course, to be an acceptable candidate to the AES process, we would expect a candidate algorithm to pass such statistical testing.

In the NIST report [3] two of the five finalists, *RC6* and *Twofish*, were initially singled out as having some non-random behavior. Later testing established that these effects were in fact consistent with both algorithms having perfectly reasonable behavior and the test failures were deemed a "statistical" effect.

The purpose of this note is to report on some additional testing of the RC6 algorithm [2] that we performed that seemed to confirm this view. While we would have been very surprised to find any true statistical weakness in the behavior of RC6, we felt that any lingering suspicion among the cryptographic community might best be dispelled by thorough testing and extensive replication of the NIST experiments.

2 NIST's methodology

The NIST document [3] is unfortunately a little unclear with regards to details at times, and as a consequence much of the material for this section has been obtained after much consultation with the author of the NIST report. We are therefore extremely grateful for Juan Soto's continued patience in answering our numerous emails!

For each algorithm, nine different sets of data were generated. We will denote these by D_1, …, D_9. Each of these sets of data was then tested using a set of statistical tests that were based around 15 basic tests. The actual number of tests applied varies between 59 and 187 and depends on the data sets under consideration. Since these details are of little importance with regards to this note, we will denote the number of tests by a and list the tests as T_1, …, T_a.

It is important to realize, that in testing a data set D_i (which might be considered as one enormously long sequence) the data set was divided into a set of 128, 300, or 384 sub-sequences. Once again, the exact number depends on the data set under consideration. During testing with test T_j, the data set D_i is generated and used to provide b sequences $S_{i,1}, \ldots, S_{i,b}$. Each run of the test T_j on each sequence provides a pass/fail decision (where this pass/fail test is made at the 1% significance level) and the overall number of pass/fail tests for the b sequences is summed. Provided the total falls under some bound that is given in [3], the data set D_i is deemed to have passed the statistical test T_j.

It is not the purpose of this note to comment on the suitability of the tests that were performed, or to comment on the style of analysis [1]. Instead, we note from [3] that there are three combinations of data set and test for which RC6 was initially thought to display non-random behavior. Here we describe the results of additional and extensive testing of RC6 with regards to these particular test and data sets.

3 NIST results on three specific tests

The three potentially anomalous results reported for RC6 were for the following data sets and tests.

1. Three templates in the *non-periodic template* test for the data set described as *128-bit key avalanche*. The templates were `01011011`, `010111011`, and `110001010`.
2. The *frequency test* for the data set described as *high density plaintext*.
3. The *cumulative sum (forward) test* for the data set described as *high density plaintext*.

3.1 The 128-bit key avalanche data set

The data sets D_i is constructed as follows. $24,576$ keys are chosen at random. For each key, the all-zero plaintext is encrypted with one of the 128 perturbed keys where a perturbed key is the base key with one of the 128 bits flipped. The resultant ciphertext for the perturbed key is then xored with the ciphertext that results from encryption using the base key.

Thus, each of $24,576$ keys yields 128 perturbed keys with which 128 bits (one output block) are generated. In total there are $24,576 \times 2^{14}$ bits that are generated to form the data set.

To perform the test, the data set D_i is considered as 384 separate sequences $S_{i,j}$ $(1 \leq j \leq 384)$. Each sequence $S_{i,j}$ consists of $64 \times 128 \times 128 = 2^{20}$ bits. The specified test is applied to all these sequences, and if there are more than nine failures[3] a note is made.

[3] See Section 5 for more detailed discussion on this issue.

The non-periodic template test. This test is made to assess whether each template occurs with the expected frequency in a test sequence. To apply the test, NIST starts with one of the 384 sequences $S_{i,j}$ constructed from data set D_i.

Each sequence $S_{i,j}$ of 2^{20} bits is divided into eight sequences of 2^{17} bits. For each of these eight subsequences we count the number of occurrences of the given template. This gives us eight scores which we denote by $f_1, \ldots, f_8$. According to NIST, for each of the eight subsequences, the expected mean and variance for the number of occurrences of a given template of length m are $\mu = \frac{2^{17}-m+1}{2^m}$ and $\sigma^2 = 2^{17}(\frac{1}{2^m} - \frac{2m-1}{2^{2m}})$. The following statistic is calculated

$$\chi^2 = \frac{\sum_{i=1}^{8}(f_i - \mu)^2}{\sigma^2}$$

and at the 1% significance level, the sequence is judged to have "failed" the test if the statistic has a value of 20.09 or more.

3.2 The high density plaintext data set

This data set is constructed as follows. 128 keys are chosen at random. For each key, the all-one plaintext is encrypted in ECB mode, as are the 128 plaintext blocks with a single zero, and the 8,128 plaintext blocks with two zeroes. For each key, this gives 8,257 blocks of 128 bits each.

NIST performs a given test on these 128 sequences $S_{i,0}, \ldots, S_{i,127}$ each of $1,056,896$ bits. Each sequence yields a pass/fail result at the 1% significance level, and if the number of failures is greater than four a note is made.

The frequency test. This test is made to assess whether the number of ones in a sequence is that expected from a truly random sequence. Consider a set of b sequences $S_{i,0}, \ldots, S_{i,b-1}$ for some data set D_i of length n where $n \geq 2^{20}$. Compute v_j $(0 \leq j \leq b-1)$, the absolute value of the difference between the number of ones and zeros in sequence $S_{i,j}$. If $\frac{v_j}{\sqrt{n}} \geq 2.575$ we reject the sequence. For a random sequence, the probability that strictly more than five sequences in a data set are rejected is 0.00192.

The cumulative sum (forward) test. This test is made to assess whether there are an undue number of ones (or zeroes) during the early stages of a sequence. To do this, the data set is tested in the following way.

Denote constituent bits in each of the b sequences $S_{i,0}, \ldots, S_{i,b-1}$ of $1,056,896$ bits, as $d_{i,j,k}$ where $d_{i,j,k}$ denotes the k^{th} bit in sequence $S_{i,j}$ and $S_{i,j}$ is the j^{th} sequence from the b sequences that were constructed from data set D_i. Then for each $S_{i,j}$ we can compute the sequence of sums $Z_t = \sum_{i=0}^{t}(2d_{i,j,k} - 1)$ for $0 \leq t \leq 1,056,895$. We take $d = \max\{|Z_t|\}_{0 \leq t \leq 1,056,895}$ where d is the maximum deflection from zero. The test performed by NIST on this information is rather complicated to describe within this short note. Instead the reader is encouraged to contact NIST for more details on this test. For our own testing purposes we replicated the procedure as described to us by NIST.

4 Our results on three specific tests

In this section we describe the results of extensive testing of the three data/test situations that were described in Section 3.

1. The *frequency test* for the data set described as *high density plaintext.*

   For each iteration of this test 128 sequences are generated each consisting of 1,056,896 bits. In our experiments we repeated the test 18,100 times and found that in 33 cases strictly more than five sequences were rejected. For a truly random sequence the expected number is 34.752.

2. Three templates in the *non-periodic template* test for the data set described as *128-bit key avalanche.* The templates are `01011011`, `010111011`, and `110001010`.

   In the test performed by NIST 384 sequences of length 2^{20} bits were examined and the number of occurrences of a given template derived. We performed this test on 1,600 sequences and then performed a goodness-of-fit test to compare the number of rejected sequences observed in each test with the number of rejections that might be expected.

   Template `01011011`**.**
   In the following table we present the distribution of the number of rejected sequences that occurred in the 1600 tests we made. For each value k, we also give the expected value for the number of tests that reject k sequences. The chi-squared value is 11.72 which is less than the indicated threshold of 20.09.

number of rejected sequences	*number of tests*	*expected number of tests*
0	42	33.73
1	127	130.84
2	270	253.08
3	350	325.51
4	299	313.18
5	230	240.42
6	156	153.40
7	68	83.67
≥ 8	58	66.14

   Template `010111011`**.**
   In the following table we present the distribution of the number of rejected sequences that occurred in the 1600 tests we made. For each value k, we also give the expected value for the number of tests that reject k sequences. The chi-squared value is 7.69 which is less than the indicated threshold of 20.09.

number of rejected sequences	*number of tests*	*expected number of tests*
0	29	33.73
1	132	130.84
2	231	253.08
3	309	325.51
4	315	313.18
5	249	240.42
6	169	153.40
7	88	83.67
≥ 8	78	66.14

Template `110001010`.
In the following table we present the distribution of the number of rejected sequences that occurred in the 1600 tests we made. For each value k, we also give the expected value for the number of tests that reject k sequences. The chi-squared value is 8.53 which is less than the indicated threshold of 20.09.

number of rejected sequences	*number of tests*	*expected number of tests*
0	35	33.73
1	112	130.84
2	234	253.08
3	333	325.51
4	343	313.18
5	231	240.42
6	160	153.40
7	87	83.67
≥ 8	65	66.14

3. The *cumulative sum (forward) test* for the data set described as *high density plaintext.*

In the following table we present the distribution of the number of rejected sequences that occurred in the 3,072 tests we made. For each value k, we also give the expected value for the number of tests that reject k sequences. The chi-squared value is 7.43 which is less than the indicated required threshold of 15.09.

number of tests failed	*number of sequences*	*number expected to fail*
0	887	848.65
1	1117	1097.24
2	661	703.78
3	288	298.57
4	90	94.25
≥ 5	29	29.51

5 One point on the NIST analysis

We feel that the following observation might be useful since it pertains to the results of the non-periodic template test that were presented in [3]. We note, however, that a more recent report from NIST [4] makes similar calculations to those that we now present here.

We have described the analysis performed by NIST [3] in a test to count the occurrences of a certain template (say). For a data set to be acceptable, the number of rejected sequences is contained within another confidence interval. In the NIST tests [3] this confidence interval was constructed using a normal approximation at the significance level of 0.001. However, explicit computations are easily performed; using maple for example. This observation was also made by Murphy [1]. In the following table we present $p(n,k)$ for $n = 128, 300$, and 384, where $p(n,k)$ is the probability that *strictly* more than k among n truly random sequences reject a test. The probability for a rejection of a single test is 0.01.

n	k	p(n,k)	n	k	p(n,k)	n	k	p(n,k)
128	4	0.00961	300	8	0.00360	384	9	0.00594
128	5	0.00192	300	9	0.00102	384	10	0.00196
128	6	0.00033	300	10	0.00026	384	11	0.00060

This table suggests that the confidence intervals constructed in [3] should be amended to include 5 when $n = 128$, 9 when $n = 300$, and 10 when $n = 384$. NIST set the rejection parameters to be 4, 8, and 9 respectively. As a consequence we see that the results provided by NIST for RC6 in the non-periodic template test for templates 01011011 and 010111011 were in fact reasonable and should not have been flagged as suspect in the first place. Nevertheless, we performed NIST-style tests on all three templates and the results were given in Section 4.

6 Conclusions

Additional (and extensive) statistical testing has been performed on RC6. NIST has gone on to confirm [3] that the results for certain statistical tests on RC6 were reasonable. The results presented in this note confirm that view.

References

1. S. Murphy. The power of NIST's statistical testing of AES candidates. Preprint. January 17, 2000.
2. R.L. Rivest, M.J.B. Robshaw, R. Sidney, and Y.L. Yin. The RC6 Block Cipher. v1.1, August 20, 1998. Available at www.rsasecurity.com/rsalabs/aes/
3. J. Soto. Randomness testing of the AES candidate algorithms. NIST. Available via csrc.nist.gov.
4. J. Soto and L Bassham. Randomness testing of the AES finalist algorithms. NIST. March 21, 2000. Available via csrc.nist.gov.

This article was processed using the LaTeX macro package with LLNCS style