Scientific Issues
Jan Długosz University
in Częstochowa
Mathematics XXI (2016)
115–122
DOI http://dx.doi.org/10.16926/m.2016.21.10

ON SECURITY PROPERTIES OF RC5 CIPHER'S NON STANDARD MODIFICATIONS

ADAM KOZAKIEWICZ AND MIROSŁAW KURKOWSKI

Abstract

The RC5 algorithm is the cipher from the family of symmetric ciphers created by Ronald Rivest. Unlike other encryption algorithms, RC5 is designed this way, that a user or a security system architect can change some of its parameters. RC5 is a block cipher processing cipher-text blocks in the sequential rounds, where input of each round is the output of the previous one. In each round data is processed with usage of the key. The parameters of the cipher that can be changed are following: length of the key, length of the processed block and number of rounds. These parameters should be chosen based on the required level of security of communication.

However there are such structures in RC5 that use of them is not entirely clear from the point of view for algorithm's security. The aim of this paper is to examine how a cryptographic power of the cipher is affected by modifications to these structures.For this purpose will be used the well-known NIST tests.

1. Introduction

Ensuring the confidentiality of transmitted data is a very important problem in modern computer networks. Widely applies here cryptography, formerly used only for military purposes. The cryptography helps to solve many problems in modern computer networks, such as access control, transmission data safety or other topics related to security of computer systems.

The very important property of cryptographic algorithms is their widely understood security. One of the measures used to assess these properties are tests developed by American National Institute of Standards and Technology (NIST) [8, 9]. During the analysis of the constructed and used algorithms one can notice that, that sometimes in their design are used structures, which role is not clear especially in the aspect of their impact on the quality of the cipher. Sometimes these constructions are not based on the thorough scientific research, but on the practical experience of their architects. The importance of the introduction of these constructions one can confirm by studying quality of the cipher after making some changes to their structure. For this can be used e.g. NIST tests mentioned before. The aim of this article is to investigate security properties of some custom modifications of RC5 using NIST tests.

The structure of this paper is as follows: in the next section RC5 cipher [1, 2, 5] will be introduced and one of interesting taken so far attempts to break it by brute-force method will be discussed. In the following section will be discussed tests proposed by NIST which goal is to verify quality of cipher. In the main section of article we will show tests results of some special custom modifications of RC5 cipher. This paper will be finished by section containing a summary of research.

2. RC5 cipher

The RC5 cipher was designed in 1994 by Ron Rivest and was initially tested by RSA Laboratories [1, 2, 5]. During the design of this cipher, Rivest had in mind following aspects:

- new cipher has to be a symmetric cipher,
- new cipher should be easy to implement both in hardware and software,
- new cipher had to be fast and possible for implementation on processors working with different lengths of word,
- cipher structure had to be iterative with variable number of cycles, which should be a parameter and compromise between fastness and security,
- new cipher should work with keys of different lengths,
- new cipher should be of course a safe algorithm strongly resistant to known cryptanalytic attacks.

An important parameter of RC5 cipher is length of a word w. The length of currently processed input/output block equals $2w$ bits. The length w is used as $w = 32$ which mean,s that lengths of plaintext and ciphertext equals 64 bits. RC5 is very customizable and can use an arbitrary length of word w. For practical use it is proposed to use lengths of word: $w = 16, 32$ or 64 bits. Another parameter of the discussed cipher is the number of cycles r, on which depends also the number of key generated by procedure of main key extension. Dependency between number of keys generated per cycle t and number of cycles is shown as follows: $t = 2(r + 1)$. It is clear, that the bigger number of cycles r the bigger is safety level. However in this case the execution speed of the cipher is significantly reduced.

Apart from different length of word and different number of cycles, RC5 can operate on keys with different length. Size of key is defined in bytes with p parameter and the particular bytes of main key are recognized as $K_0, K_1, \ldots, K_{b-1}$.

The variety of values of parameters mentioned above, which could be used in practice caused the introduction of special notation to describe cipher: $RC - 5w/r/b$, where: w - size of word in bits, r - number of cycles, b - length of main key K in bytes. For example: $RC5 - 32/16/10$ means, that this cipher processes 64-bit blocks, uses 80 bits key and executes 16 cycles.

Algebraic operations used in RC5 encryption and decryption processes are:

(1) sum of two numbers in finite field $GF(2^w)$, which is denoted as summing character $+$ and reverse of this operation is marked as $-$
(2) XOR - modulo 2 sum of two integers,
(3) periodic left shift; shift of word x for y bits is marked as $x <<< y$ (reverse operation - periodic right shift is marked as $x >>> y$).

In encryption process for each cycle there are used two keys S, generated by method of extending main key, marked as S_{2i} and S_{2i+1}. It is assumed that input block with size $2w$ bits is divided into two blocks with w size, for registers A and B.

RC5 encryption process is as follows:

```
$A \leftarrow A + S_0$
$B \leftarrow B + S_1$
$for\ i = 1\ to\ r\ do$
$A \leftarrow ((A\ XOR\ B) <<< B) + S_{2i};$
$B \leftarrow ((A\ XOR\ B) <<< A) + S_{2i+1}$
```

At the beginning values of registers A and B are added in $GF(2^w)$ field to key values S_0 and S_1. Then, during next r cycles there are used: XOR of A and B contents, cyclic shifts and addition of shifted values with keys of cycle S in field $GF(2^w)$. It is important, that this shift occurs as the only one non-linear operation of this algorithm. In single cycle values of both registers are changed.

The RC5 decryption process de facto makes the inverse operations of encryption in the reverse order. Decryption algorithm can be then shown as follows:

```
**for** $i = r$ **downto** 1 **do**
    $B \leftarrow ((B - S_{2i+1}) >>> A)\ XOR\ A$
    $A \leftarrow ((A - S_{2i}) >>> B)\ XOR\ B$
    $B \leftarrow B - S_1$
    $A \leftarrow A - S_0$
```

Similarly as in most ciphers, RC5 uses the algorithm of the extension of the main key to generate different key for each cycle - iteration. For this purpose RC5 uses two numbers P_w and Q_w with size of word w.

They are defined as follows:

Date	Jun. 05	Jun. 07	Sep. 08	May 09	Sep. 11	Feb. 12	Sep. 12	Dec. 12	Sep. 13	Oct. 15	Jun. 16	Aug. 16
Percent Compl.	0.226	0.4	0.519	0.61	1.843	2.182	2.542	2.672	2.957	3.753	4.083	4.175
Time [days]	902	1574	2112	2359	3201	3362	3579	3679	3941	4691	4941	5021

Table 1. RC5 72-bit key challenge - statistics

$P_w = Odd((e-2)\cdot 2^w)$,
$Q_w = Odd((\phi-1)\cdot 2^w)$,
where: e is a base of the natural logarithm, and ϕ is a value of the gold number. Function $Odd(x)$ rounds x value up to the nearest integer number.

RC5 is a very popular cipher used in many commercial and non-commercial implementations. Its cryptographic power is confirmed both by NIST tests [8, 9] and by cryptanalysts research [3, 4, 6, 7]. Strong resistance against brute-force attack was studied by organization distributed.net, which has organized world challenge to brute break this cipher. In case of cipher with key length equal 72 the results of breaking are shown in Table below.

Several years ago an organisation *distributed.net* started a challenge of brute-force breaking of RC5 cipher. For many years a few kinds of RC5 were broken. To find secret RC5's with the 64-bit key computations took 1726 days by searching 82.77 percent of total dictionary [10, 11]. A distributed computing project currently works on RC5 72-bit RSA secret-key challenge which volunteer's machines cooperation, simply by method brute-force. The table 1 shows how it was changing since 2005 to present.

As it can be seen searching completely 72-bits dictionary is still in progress under 5%, so it is the proof this key is hard to find by brute-force method.

3. NIST tests

The need for random and pseudorandom numbers arises in many cryptographic applications such as RC5. The NIST Test Suite is a statistical package consisting of 15 tests that were developed to test the randomness of (arbitrarily long) binary sequences produced by either hardware or software based on cryptographic random or pseudorandom number generators. These tests focus on a variety of different types of non-randomness that could exist in a sequence. Some tests are decomposable into a variety of subtests. The 7 of 15 tests are:

Zastosowane testy:

- Frequency (Monobits) Test - The focus of the test is the proportion of zeroes and ones for the entire sequence. The purpose of this test is to determine whether that number of ones and zeros in a sequence are approximately the same as would be expected for a truly random sequence. The test assesses the closeness of the fraction of ones to $\frac{1}{2}$, that is, the number of ones and zeroes in a sequence should be about the same.
- Test for Frequency within a Block - The focus of the test is the proportion of zeroes and ones within M-bit blocks. The purpose of this test is to determine whether the frequency of ones is an M-bit block is approximately $\frac{M}{2}$.
- Runs Test - The focus of this test is the total number of zero and one runs in the entire sequence, where a run is an uninterrupted sequence of identical bits. A run of length k means that a run consists of exactly k identical bits and is bounded before and after with a bit of the opposite value. The purpose of the runs test is to determine whether the number of runs of ones and zeros of various lengths is as expected for a random sequence. In particular, this test determines whether the oscillation between such substrings is too fast or too slow.
- Test for the Longest Run of Ones in a Block - The focus of the test is the longest run of ones within M-bit blocks. The purpose of this test is to determine whether the length of the longest run of ones within the tested sequence is consistent with the length of the longest run of ones that would be expected in a random sequence. Note, that an irregularity in the expected length of the longest run of ones implies that there is also an irregularity in the expected length of the longest run of zeroes. Long runs of zeroes were not evaluated separately due to a concern about statistical independence among the tests.
- Random Binary Matrix Rank Test - The focus of the test is the rank of disjoint sub-matrices of the entire sequence. The purpose of this test is to check for linear dependence among fixed length substrings of the original sequence.
- Discrete Fourier Transform (Spectral) Test - The focus of this test is the peak heights in the discrete Fast Fourier Transform. The purpose of this test is to detect periodic features (i.e., repetitive patterns that are near each other) in the tested sequence that would indicate a deviation from the assumption of randomness.

- Non-overlapping (Aperiodic) Template Matching Test - The focus of this test is the number of occurrences of pre-defined target substrings. The purpose of this test is to reject sequences that exhibit too many occurrences of a given non-periodic (aperiodic) pattern. For this test and for the Overlapping Template Matching test, an m-bit window is used to search for a specific m-bit pattern. If the pattern is not found, the window slides one bit position. For this test, when the pattern is found, the window is reset to the bit after the found pattern, and the search resumes.

4. Experiment results

As mentioned in the introduction the purpose of this paper is to study quality of RC5 cipher with custom, non standard modifications. Firstly it was changed number of rotation in encryption algorithm and the values of constants P and Q in algorithm for generating sub-keys. The use of particular number of rotations and mentioned before P and Q values in RC5 is not proved by any cryptanalysis thesis. Research objectives are experimental verification if changes of those cipher parameters cause a reduction of its cryptographic power related to NIST tests.

As stated in the introduction for research was used own implementation of cipher made according to original specification. Standard output of working RC5 algorithm application has been used as base for further comparison. After analysing 15 different statistical tests it could be concluded that most of them (14 of 15, except for Entropy Test) succeeded.

Secondly, the experiment was repeated with increased input data file and increased data probing from 25 to 100 bitstreams. Mean success ratio for 10 tests reached 95%, where Test for Frequency within a Block and Test for the Longest Run of Ones in a Block gave 100% result, despite analyzed ciphertext beeing longest. Other 4 test succeed as well up to 93%. Only Entropy Test gave 68%.

RC5 algorithm was tested with use of special modifications described before. Results for different test cases deviated no more than 1%. For this reason and considering that total content of this paper should be sufficient in the description below we will show general results with no particular discussion about each case.

Frequency Test each time gave results above 98%, for modification $P, Q = 0x00000000$ gave 99%, and for $P, Q = 0xFFFFFFFF$ 100%. Test for Frequency within a Block gave results not less than 99%. As the result of Cumulative Sum Test minimum value was 97% only for custom modification with removed rotation. Similar only for this modification (removed rotation) score of Runs Test was minimal 97%. Run Test for others gave even two times 100%. Test for the Longest Run of Ones in a Block underperformed slightly in case with no rotation, because equaled 95%, however for other cases results were between $99 - 100\%$. Random Binary Matrix Rank Test reached maximum limit 100% for custom modification without rotation and minimum was 98% for probe encrypted by RC5 with 6-times rotation. Discrete Fourier Transform Test had the worst score only for modification with 6-times rotation, for others score was $99 - 100\%$. It should be also mention about Overlapping Template Matching Test (based on 0 I 1 given by default) made for each probe when results were above 99% (only for $P, Q = 0xFFFFFFFF$: 98%). Non-overlapping Template Matching Tests (147 samples for each) also succeeded, minimum value was 78% and maximum value was 100%, however some of those tests went even better for custom modifications than for standard RC5 algorithm without any. Only Approximate Entropy Test gave worse results for each case (for $P, Q = 0xFFFFFFFF$: 56%; for $P, Q = 0x00000000$: 58%; $P, Q = 0xFFFFFFFF$: 64%). Exception was results of Entropy Tests for custom modification with no rotation, it was 0% for bigger input file and 100% for smaller input file. ($C1 - C9 = 0, C10 = 25/25$).

Serial Test and Approximate Entropy Test were slightly worse but could be considered successful, however for bigger input file and custom modification without rotation there was exception: 2 Serial Tests gave 0%, 9%).

To sum up, results of NIST tests for customized modifications RC5 cipher succeeded, regardless of which kind of modification was made. For smaller file, when probe was too small, the tests did not start. Only Entropy Tests (or Serial Tests) went worse for each case, regardless of size of input file yet still better than results of original RC5 algorithm.

5. Final remarks

As shown in the results about discussed custom changes of the cipher, they have not negative impact on its cryptographic power. Custom changes such as different rotation times or constants value for generate round keys did not affect NIST tests score, so they can not be found as unnecessary. The discussed special constructions that appear in RC5 are unreasonable from the point of view for cipher security and properties checked by the tests. The other reasons for introducing them are not known for us. Further research can be related withinvestigation to find other proofs about introduction of included constructions in cipher algorithm.

References

[1] Rivest, R. L., *The RC5 Encryption Algorithm*, Fast Software Encryption: Second International Workshop. Leuven, Belgium, 14-16 December 1994, Proceedings, vol. 1008 of LNCS, pp. 86–96, Springer Verlag, 1995.

[2] Menezes A. J. , van Oorschot P. C., Vanstone S. A., *Handbook of Applied Cryptography*, CRC Press, 1996.

[3] Knudsen L. R., Meier W., *Differential cryptanalysis of RC5*, European Transactions on Telecommunications 8(5), pp. 445-454, 1997. DOI: 10.1002/ett.4460080503

[4] Biryukov A., Kushilevitz E., *Improved Cryptanalysis of RC5*, EUROCRYPT 1998, vol. 1403 of LNCS, pp. 85-99, Springer Verlag,1998.

[5] Rivest, R. L., *Block Encryption Algorithm With Data Dependent Rotation*, U.S. Patent 5,724,428, issued on 3 March 1998.

[6] Knudsen L. R., Meier W., *Improved Differential Attacks on RC5*, Advances in Cryptology — CRYPTO '96, vol. 1109 of LNCS, pp. 216-228, Springer Verlag, 2001.

[7] Hasan M., Al-Shalabi H., *Modified Cryptanalysis of RC5.* Int. Arab J. Inf. Technol. 3(4), pp. 299-302, 2006.

[8] *http://csrc.nist.gov/groups/ST/toolkit/rng/documentation_ software.html*

[9] *http://csrc.nist.gov/groups/ST/toolkit/rng/stats_ tests.html*

[10] *http://www.distributed.net/RC5*

[11] *http://stats.distributed.net/projects.php?project_ id=8*

Received: September 2016

Adam Kozakiewicz
Military Communication Institute,
ul. Warszawska 22A, 05-130 Zegrze Południowe, Poland
E-mail address: a.kozakiewicz@wil.waw.pl

Mirosław Kurkowski
Cardinal Stefan Wyszyński University,
Institute of Computer Science,
ul. Dewajtis 5, 01-815 Warsaw, Poland
E-mail address: m.kurkowski@uksw.edu.pl