Improving the RC5RA Algorithm’s Crypto Resistance for Embedded Computers

Andrii Sahun[1], Vladyslav Khaidurov[2], and Pavlo Gikalo[3]

[1] *National University of Life and Environmental Sciences of Ukraine, 15 Heroyiv Oborony str., 03041, Kyiv, Ukraine*
[2] *Institute of Engineering Thermophysics of NAS of Ukraine, 2a Mariyi Kapnist str., 03057, Kyiv, Ukraine*
[3] *National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute,” 37 Peremohy str., 03056, Kyiv, Ukraine*

Abstract

An approach to increase the cryptographic stability of the RC5RA classical cryptographic algorithm. The proposed approach does not increase the computational complexity of the RC5RA algorithm due to the fact that it does not involve increasing the encryption rounds, key or block length. The approach to crypto resistance improvement of this algorithm is based on the use of nonlinear round shift functions. These functions are continuous throughout the entire range of their existence. The obtained model of the RC5RA crypto system demonstrates resistance to encryption of encrypted data up to 210 times (using differential analysis) at 14 rounds of encryption, at 12 rounds—the difference in crypto resistance of the modified RC5RA (unmodified version) is 2^4. The modified model of the RC5RA crypto system does not show an increase in computational time (compared to the base RC5RA). In the obtained RC5RA crypto system there are no collisions and statistical correlations between the blocks of incoming messages and outgoing blocks.

Keywords

Block cipher, symmetric encryption, nonlinear shift function, RC5RA, symmetric block cipher cryptanalysis.

1. Introduction

Data encryption in computer network channels can be implemented at any of the seven levels of the OSI model [1]. Encryption is more often implemented either at the upper (application) or lower (channel, network) levels of the OSI model. In addition to "useful" information traffic, information about the routing of the message, information about the routing protocol are also encrypted. In this case, the network switch must decrypt the data stream in order to process it correctly. Then the switch encrypts the traffic again for transmission to another network switch. For this reason, «Internet of Things» networks, especially those based on low-power controllers (Arduino, NodeMCU, ESP32), use special methods of information security to protect network traffic. Firewalls are most often used for this purpose. It should be noted that channel encryption in such cases is an effective means of information protection [2].

Despite the high efficiency, channel encryption has disadvantages:

– the cost of implementing encryption at the channel level increases sharply with increasing network size;

– the data must be encrypted each time it is transmitted over a network channel.

Trends in the development of cryptographic protection (for example, the hardware implementation in the computing cores of the microprocessor of the AES encryption algorithm [3]) indicate the prospects for the implementation of block XOR-ciphers. Thus, a large number of means of cryptographic computer protection is implemented in the form of hardware units or devices.

CPITS-II-2021: Cybersecurity Providing in Information and Telecommunication Systems, October 26, 2021, Kyiv, Ukraine

EMAIL: avd29@ukr.net (A. Sahun); allif0111@gmail.com (V. Khaidurov); p.gikalo@gmail.com (P. Gikalo)
ORCID: 0000-0002-5151-9203 (A. Sahun); 0000-0002-4805-8880 (V. Khaidurov); 0000-0002-8058-8267 (P. Gikalo)

Hardware encryption has a higher speed. For example, the RC5 cryptographic algorithm consists of a large number of modulus arithmetic operations that are performed on plaintext bits. However, modern microcontrollers are practically not adapted to perform efficient bit crypto transformations. The creators of the RC5 cryptographic algorithm provided that it could be easily implemented by both hardware and software methods, which allows to protect network traffic for Internet of Things (IoT) technology. Although at the moment firewalls and some other means and technologies of data protection are more often used for this [4–7]. The algorithms of the RC5 family provide for the division of the message into a certain number of parts of a fixed size. Each of these parts undergoes a separate encryption procedure. This simplifies the encryption task. The given practical capabilities of the RC5 algorithm can confirm that the block symmetric RC5 encryption algorithm has prospects in IoT. It provides prospects for enhancing the quality of encryption and reducing the computational load on the computing mechanisms of microcontrollers [3, 8].

In their work [3], the authors of the RC5 crypto algorithm note that the proposed algorithm can be easily implemented in hardware. At the same time, the ways of enhancing its cryptographic power without increasing the computational complexity are also proposed [8]. This feature of RC5 is important in its application and is a convenient basis for modification [3]. Although, the classical algorithm is built on one shift function, it can be modified by using several sequentially nonlinear shift functions [3, 9].

2. Analysis of the Principle of Operation and Parameters of the RC5 Algorithm

Some of the main parameters of the RC5 algorithm are variable parameters [3, 8]. As a basic algorithm for modification, we choose its RC5RA variation [10]. In it, the cyclic shift occurs by a variable number of bits, which depends on the algorithm round number, determined by a function f (). This function processes all bits of another sub block as an input value. The scheme of cryptographic transformation in rounds of the RC5RA algorithm is shown in Figure 1. In the algorithms of the RC5RA family, in addition to the secret key, there are some others, namely:

– the size of the word w (in bits). The algorithm encrypts blocks of two words (hereinafter referred to as A and B respectively). Valid values of w are natural numbers 16, 32 or 64. The recommended word size is 32 bits;

– the number of rounds of the R-algorithm (any integer in the range from 0 to 255 inclusive);

– secret key size b (in bytes) – variable value (any integer in the range from 0 to 255 inclusive). When encrypting for two blocks A and B in the binary representation, the RC5 algorithm is executed in such a way that before the first encryption round the operations of superimposing the extended key S on the encrypted data according to expression (1) are performed.

The maximum RC5 key size for RC5 family algorithms is 2040 bits. An example of the formation of the extended key S is given in [1]. The RC5RA algorithm performs a cryptographic conversion of the kind:

$$A_{i+1} = \left((A_i \oplus B_i) << f(\cdot)\right) + S_{2i},\ \ B_{i+1} = \left((B_i \oplus A_i) << f(\cdot)\right) + S_{2i+1}. \qquad (1)$$

The proposed approach to improvement involves not so much a modification of the algorithm, but an increase in its crypto resistance by choosing the nonlinear shift functions used in (1). This function, in turn, is the basis for modifying the algorithm. Obviously, the main point for improving the RC5RA is the choice of nonlinear functions of the form $f(K,r)$, where the K – round key and r – number of the round. The using of the following functions is proposed as nonlinear functions:

1) the first shift function:

$$f(K,r) = \left| r + \left[w\, sin\left(wr \sum_{s=1}^{w} L_{bit_s} \right) \right] \right| mod\, w, \qquad (2)$$

where $m = r^2 mod\, w$;
r – round number;
w – the length of half of the coded block;

$[x]$ – an integer part of the x number;

L_m – is the length of the initial message block (plaintext) in the encryption round;

L_{bit} – a binary representation of the symbol Lm encoded by the symbol of the word K, the length of which is w in a bit representation.

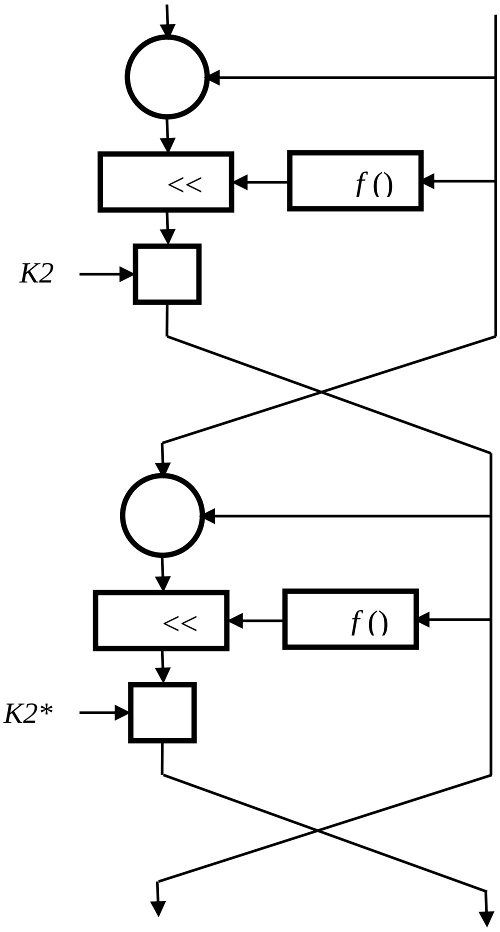


Figure 1: Scheme of cryptographic transformation in RC5RA algorithm rounds

2) the second offset function:

$$f(K,r) = \left[w\, exp\left(-\left| \frac{w}{10} sin\left(wr \sum_{s=1}^{w} L_{bits_{m_s}} \right) \right| \right) \right] mod\, w. \qquad (3)$$

3) the third offset function:

$$f(K,r) = \left[3r \left| th\left(\frac{w}{10} sin\left(wr \sum_{s=1}^{w} L_{bits_{m_s}} \right) \right) \right| \right] mod\, w. \qquad (4)$$

4) the fourth offset function:

$$f(K,r) = \left(\left| rw \sum_{s=1}^{w} L_{bits_{m_s}} + \left[rw\, cos\left(3 \sum_{s=1}^{w} L_{bits_{m_s}} + 2r \right) \right] \right| \right) mod\, w. \qquad (5)$$

5) the fifth offset function:

$$f(K,r) = \left(\left|\sum_{s=1}^{w} L_{bits_{m_s}} + \left[log_2\left(1 + rw\left|\sum_{s=1}^{w} L_{bits_{m_s}}\right|\right)\right] mod\, r\right|\right) mod\, w. \quad (6)$$

The given nonlinear shift functions (2)–(6) have a range of admissible values for the whole period of their existence. This is necessary to form the coefficients in the encryption rounds.

3. Formation of Test Samples for the RC5RA Algorithm

The implementation of a modified algorithm requires test data for encryption. When forming test samples, we will assume that for the RC5 cipher or RC5RA modification there are no confirmed direct relationships between the sizes/multiplicities of the sample text files with open text and the obtained encryption result. And all the more there is no influence of these parameters on crypto resistance of the received encrypted text or emergence of collisions. Files containing only integer data were used for test samples. When encoding graphics and texts using known and current coding systems, we obtain a file in the form of a sequence of integers. Thus, text test datasets are 81 to 1968 kilobytes long, and graphical data sets are 351 to 3412 kilobytes long. Text data are files with English characters. Graphic data is a set of simple images of standard formats, such as the jpg format. Encryption keys are generated using a pseudo-random number generator with modular arithmetic. In this case, the keys are formed immediately in the bit representation. 10 keys are generated for each input data set.

4. Application of a RC5RA Algorithm's Modification

A software application in the Matlab modeling environment was created to determine the parameters of the obtained RC5RA modification. The program interface provides the ability to change the encryption parameters of the modification of the RC5 algorithm. To test the obtained modification of the RC5RA algorithm and obtain the results of its work, identical data of key parameters, encryption rounds and function numbers of each series of test samples were used. A sequence of different bits (16, 32, 64 bits) was used as the encryption key. The key sequence was generated by a standard built-in of the Matlab programming language congruent pseudo-random number generator.

The results obtained for modifying the RC5RA using the proposed shift functions (2)–(6) for different parameters of the algorithm are shown in Figures 2–5.

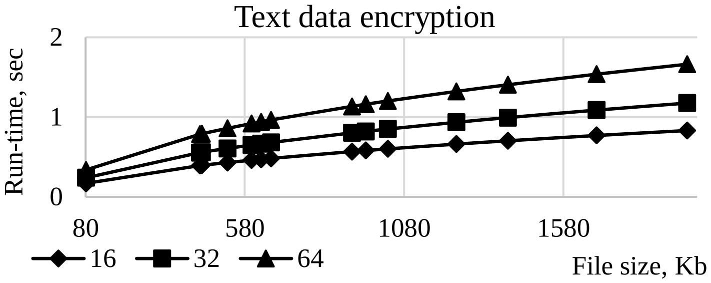


Figure 2: Text data encryption time at r = 16 and k = 16, 32, 64 bits

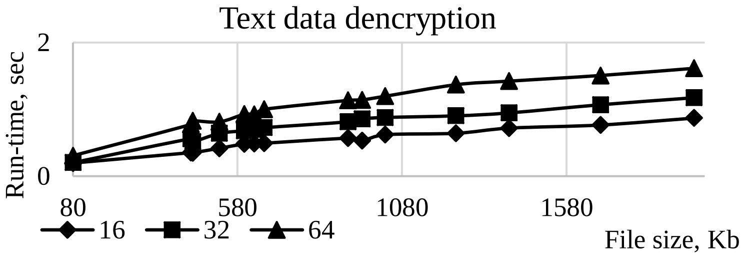


Figure 3: Text data decryption time at r = 16 and k = 16, 32, 64 bits

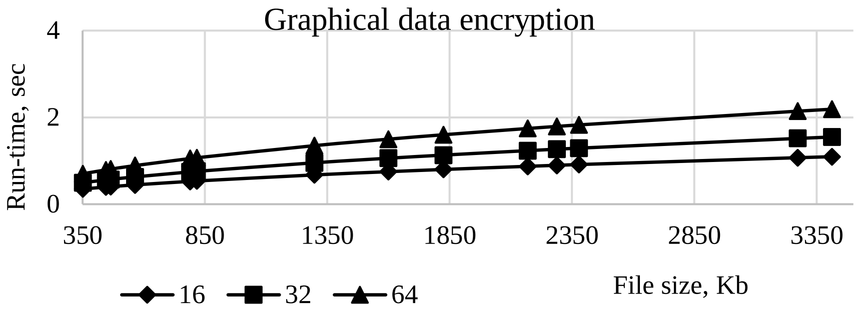


Figure 4: Graphic data encryption time at r = 16 and k = 16, 32, 64 bits

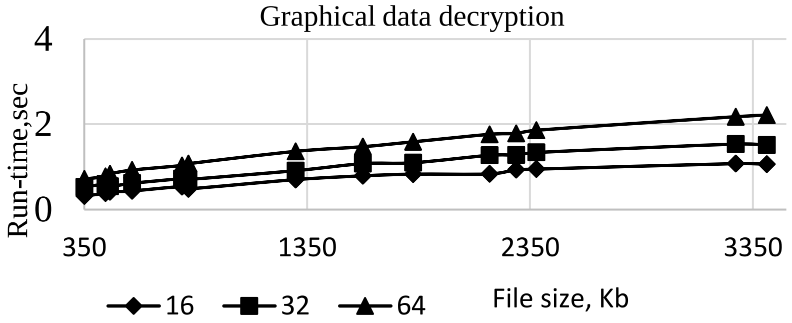


Figure 5: Graphic data decryption time at r = 16 and k = 16, 32, 64 bits

The data in Figures 2–5 are obtained with the classical round shift and the use of the five nonlinear functions proposed above. The results of the work obtained as a result of processing the test samples by the modified RC5RA algorithm are given in Tables 1, 2. Coincidences appear on all used test samples.

Table 1
The time of encryption and decryption of text files with the number of rounds r = 16

File size, Kbytes	Program run-time, seconds: Encryption/decryption w=16/w=16	File size, Kbytes Encryption/decryption w=16/w=16	Program run-time, seconds: Encryption/decryption w=16/w=16
81	0,17 / 0,20	0,24 / 0,20	0,34 / 0,31
439	0,39 / 0,35	0,55 / 0,56	0,78 / 0,78
445	0,40 / 0,35	0,56 / 0,52	0,79 / 0,83
525	0,43 / 0,42	0,61 / 0,64	0,86 / 0,81
601	0,46 / 0,48	0,65 / 0,68	0,92 / 0,93
631	0,47 / 0,49	0,67 / 0,70	0,94 / 0,92
662	0,48 / 0,49	0,68 / 0,73	0,96 / 1,00
916	0,57 / 0,57	0,80 / 0,81	1,13 / 1,13
959	0,58 / 0,53	0,82 / 0,86	1,16 / 1,14
1029	0,60 / 0,62	0,85 / 0,88	1,20 / 1,20
1244	0,66 / 0,64	0,93 / 0,90	1,32 / 1,37
1406	0,70 / 0,72	0,99 / 0,95	1,40 / 1,42
1684	0,77 / 0,77	1,09 / 1,07	1,54 / 1,51
1968	0,83 / 0,87	1,17 / 1,17	1,66 / 1,61

Table 2
Encryption and decryption time of graphic file modules at the number of rounds r = 16

File size, Kbytes	Program run-time, seconds: Encryption/decryption w=16/w=16	File size, Kbytes Encryption/decryption w=16/w=16	Program run-time, seconds: Encryption/decryption w=16/w=16
351	0,35 / 0,32	0,50 / 0,53	0,70 / 0,72
446	0,40 / 0,39	0,56 / 0,60	0,79 / 0,78
465	0,40 / 0,41	0,57 / 0,54	0,81 / 0,84
565	0,45 / 0,44	0,63 / 0,62	0,89 / 0,93
789	0,53 / 0,54	0,74 / 0,73	1,05 / 1,03
817	0,54 / 0,49	0,76 / 0,71	1,07 / 1,08
1298	0,67 / 0,70	0,95 / 0,91	1,35 / 1,37
1600	0,75 / 0,79	1,06 / 1,08	1,50 / 1,47
1825	0,80 / 0,83	1,13 / 1,10	1,60 / 1,59
2169	0,87 / 0,84	1,23 / 1,27	1,74 / 1,77
2288	0,90 / 0,94	1,27 / 1,28	1,79 / 1,79
2379	0,91 / 0,95	1,29 / 1,34	1,83 / 1,86
3273	1,07 / 1,08	1,52 / 1,54	2,14 / 2,18
3412	1,09 / 1,06	1,55 / 1,52	2,19 / 2,22

The Tables 1, 2 show the obtained time parameters of the modified version of the RC5RA algorithm in the mode of encryption and decryption of graphic information with the number of rounds *r* = 16 and different values of the *w*-parameter.

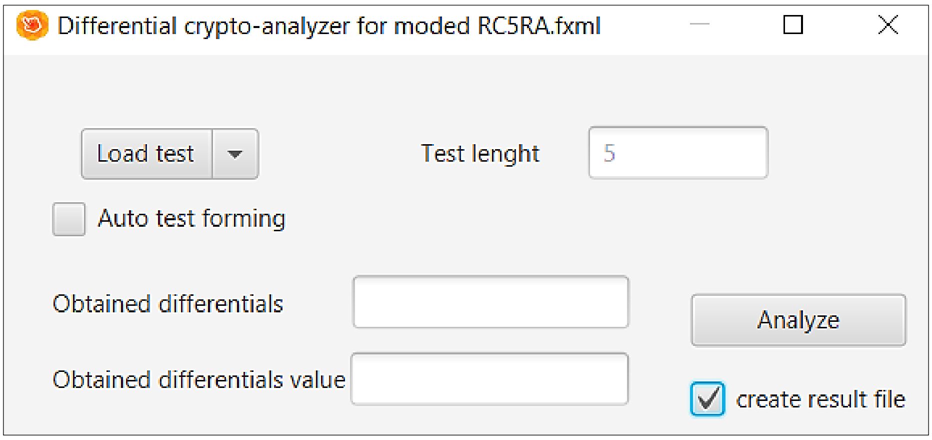
5. Cryptanalysis of the Resulting RC5RA Modification

In order to determine the value of crypto resistance of the obtained modification of the RC5RA crypto algorithm using the above-mentioned shift functions, a classical differential analysis was used [11–13].

The test data set (texts) consisted of the first 5 characters of the Latin alphabet. These specially prepared texts contained 2 and 4 consecutive symbols. A "differential" was calculated for different pairs of texts from this sample. On the basis of the received "differential" the estimation of "differential" of other pairs of encrypted texts was carried out. The model on the basis of which the cryptanalysis of the received texts was carried has the form of (7):

$$S(RC5RA) = \frac{1}{\left(1+\frac{2^{96}}{2^{72}}\right)\cdot\frac{1}{2^{11}}} + \frac{1}{\left(1+\frac{2^{96}}{2^{72}}\cdot 2\right)\left(1+\frac{2^{96}}{2^{92}}\cdot 2^{3}\right)} + \frac{1}{\left(1+\frac{2^{92}}{2^{88}}\right)\cdot\frac{1}{2^{3}}} \approx \frac{1}{5}. \quad (7)$$

The layout of the software interface, which performs cryptanalysis of the obtained modification of RC5RA is shown in Figure 6.



Test lenght 5

□ Auto test forming

Obtained differentials

Obtained differentials value

☑ create result file

Figure 6: Layout of the software interface, which performs cryptanalysis of the obtained RC5RA modification

Cryptanalysis was performed with the number of rounds of RC5RA modifications from 10 to 14, as shown in Table 3. The number of plain texts required to hack the modified RC5RA algorithm is estimated.

Table 3
Number of input (simple) texts for cryptanalysis

Number of rounds / Number of nonlinear function	r=10	r=11	r=12	r=13	r=14
The first function	2^{46}	2^{50}	2^{56}	2^{62}	2^{80}
The second function	2^{46}	2^{52}	2^{58}	2^{66}	2^{88}
The third function	2^{46}	2^{54}	2^{60}	2^{66}	2^{90}
The fourth function	2^{46}	2^{54}	2^{62}	2^{66}	2^{92}
The fifth function	2^{46}	2^{56}	2^{64}	2^{68}	2^{96}

The result of cryptanalysis of the encrypted text (Table 3) is considered successful when the evaluation of the "differential" receives a limit value of 25%. This means that with a probability of 0.25 with such structure of data, you can open half of the key or the whole key. Based on the results shown in Table 3 (by number of texts), when implementing this algorithm on low-power microcontrollers in IoT technology, the recommended number of rounds should not be less than 11. Otherwise, the crypto resistance index will not be satisfactory [13].

6. Conclusion

As a result of the analysis and generalization, the answer to the question of the influence of the nature of the round shift in RC5 on increase of the crypto resistance of this crypto algorithm was partially obtained [11].

As can be seen in Table 3, the best values of crypto resistance are demonstrated by the modified algorithm for those functions whose variance of output values is most homogeneous. Selection of a "strong" nonlinear shift function for the RC5RA improvement reduces the number of rounds to 10, while the encrypted message will be well protected from differential cryptanalysis. At the same time, the computational complexity of the algorithm implementation remains comparable to the classic version of RC5RA.

7. References

[1] ITU-T Recommendations. ITU-T X.200, Committed to Connecting the World, 1994. URL: https://www.itu.int/rec/T-REC-X.200-199407-I.

[2] R. L. Rivest, The RC5 Encryption Algorithm, in: Proceedings of the Second International Workshop on Fast Software Encryption (FSE) 1994, pp. 86–96. URL: http://people.csail.mit.edu/rivest/Rivest-rc5rev.pdf.

[3] S. Gueron, White Paper: Intel Advanced Encryption Standard (AES), New Instructions Set, Revision 3.01, 2012. URL: https://software.intel.com/content/dam/develop/external/us/en/documents/aes-wp-2012-09-22-v01-165683.pdf.

[4] A. Blozva, et al., IoT Devices Integration and Protection in available Infrastructure of a University computer Network, Journal of Theoretical and Applied Information Technology 99(08) (2021) 1820–1830.

[5] V. Lakhno, et al., The information technologies in the tasks of planning of smart city development, Journal of Theoretical and Applied Information Technology 99(14) (2021) 3645–3662.

[6] V. Lakhno, et al., Methodology for assessing the effectiveness of measures aimed at ensuring information security of the object of informatization, Journal of Theoretical and Applied Information Technology 99(14) (2021) 3417–3427.

[7] V. Lakhno, et al., Development of a Model for Choosing Strategies for Investing in Information Security, Eastern-European Journal of Enterprise Technologies 2(3) (2021) 43–51. doi:10.15587/1729–4061.2021.228313.

[8] O. Elkeelan, A. Olabisi, Performance Comparisons, Design, and Implementation of RC5 Symmetric Encryption Core using Reconfigurable Hardware, Journal of Computers 3(3) (2008) 48–55.

[9] T. Zhovnovach, et al., Modification of RC5 cryptoalgorythm for electronic data encryption systems, Ukrainian Scientific Journal of Information Security 25(3) (2019) 138–143. doi:25.10.18372/2225-5036.25.14458.

[10] S. Panasenko, Algorithms are encrypted, Special reference, BHV, Sankt-Peterburh, 2009. (In Russian).

[11] A. Biryukov, E. Kushilevitz, Improved cryptanalysis of RC5, in: K. Nyberg (Eds.), volume 1403 of Lecture Notes in Computer Science. Advances in Cryptology – Eurocrypt, Springer, Berlin, Heidelberg, 1998. doi:10.1007/BFb0054119.

[12] B. S. Kaliski, Y. L. Yin, On Differential and Linear Cryptanalysis of the RC5 Encryption Algorithm, in: D. Coppersmith (Eds.), vol. 963 of Lecture Notes in Computer Science. Advances in Cryptology, CRYPTO, Springer Verlag, Berlin, Heidelberg, 1995. doi:10.1007/3-540-44750-4_14.

[13] L. R. Knudsen, W. Meier, Differential Cryptanalysis of RC5, European Transactions on Telecommunications 8(5) (1997) 445–454.