Problems of Information Transmission, Vol. 41, No. 4, 2005, pp. 385–394. Translated from Problemy Peredachi Informatsii, No. 4, 2005, pp. 97–107.
Original Russian Text Copyright © 2005 by Ryabko, Monarev, Shokin.

A New Type of Attacks on Block Ciphers[1]

B. Ya. Ryabko⋆, V. A. Monarev⋆⋆, and Yu. I. Shokin⋆⋆

⋆*Siberian State University of Telecommunication and Information Science*
Institute of Computational Technologies, Siberian Branch of the RAS, Novosibirsk
boris@ryabko.net
⋆⋆*Institute of Computational Technologies, Siberian Branch of the RAS, Novosibirsk*
vitox@gorodok.net shokin@ict.nsc.ru

Received December 7, 2004

Abstract—A new attack (called "gradient statistical") on block ciphers is suggested and experimentally investigated. We demonstrate the possibility of applying it to ciphers for which no attacks are known except for the exhaustive key search.

1. INTRODUCTION

Block secret-key ciphers are very widely adopted in information (transmission and storage) protection systems, and therefore some researchers call them a "workhorse" of cryptography. Because of this wide practical use, problems of both designing reliable block ciphers and finding effective cryptological attacks on these ciphers (i.e., methods of determining a secret key of a cipher based on experiments with encrypted messages) become issues of the day. Researches in these fields are made in parallel and often by the same specialists, and inventing a new attack leads as a rule to finding new ciphers resistant to this attack. Note that of interest for cryptography are attacks that are less time-consuming than the direct exhaustive key search. For example, if an attack requires examination of 2^{200} keys instead of, say, 2^{250} keys necessary for the exhaustive search, then the attack is of interest for cryptology [1]. Reviews of present-day block ciphers, methods of their construction and analysis, and various types of attacks can be found in many works, e.g., in [1–5], where some tens of modern ciphers and attacks are described. Numerous national and international programs and contests aimed at designing reliable block ciphers, held nowadays in the USA, European Community, Japan, and Korea (see a review in [2]), give evidence of urgency of the problem.

Most block ciphers can be described as functions defined on the set of all binary words of length $\ell + k$ and taking values in the set of binary words of length ℓ, where ℓ is the length of an enciphered word (or block) and k is the length of a (secret) key. In modern ciphers, the block length is usually either 128 or 64 bits, and the key length for different ciphers (and different modes of using a cipher) ranges from several tens to several thousand bits. For instance, in the AES cipher, winner of the 21st Century Block Cipher contest held in the USA in 1999–2001, the block length ℓ is 128 bits, and the key length may have three values: 128, 196, and 256 bits. In popular ciphers R5 and RC6, suggested by R. Rivest, the block length can be 32, 64, or 128 bits, and the key length in various versions varies from 64 to several thousand bits. It should be noted that RC5 and RC6 have a very simple description and, apparently, have therefore become among the most popular cryptanalysis objects. A brief description of RC5 is given in the Appendix.

[1] Supported in part by the Russian Foundation for Basic Research, project no. 03-01-00495, INTAS, Grant 00-738, and the Royal Society, UK, Research Grant no. 15995.

In [1, 5–7], cryptanalysis results for RC5 and RC6 were presented, and a conclusion was made about high resistance of these ciphers to all attacks described in the literature.

In many (or even all) modern block ciphers, encryption process is divided into a series of comparatively simple stages, often called rounds. In the course of each new round, data obtained in the preceding stage is encrypted using the so-called round key. All these stages are labeled in the RC5 cipher (see the Appendix). Here we should make a remark concerning the terminology: traditionally, in the description of RC5, the term *half-round* is used; two half-rounds constitute a round [3, 7]. Therefore, when proposing an attack, we speak about a round; when analyzing its applicability to RC5, about a half-round (we hope this would not cause ambiguity).

In RC5, RC6, and many other ciphers, the number of rounds is a parameter, and often cryptanalists study the "resistance" of a cipher as a function of the number of rounds. One of the goals of this analysis is to find the number of rounds ensuring high reliability of the cipher.

In the decryption process, the same rounds with the same round keys are in fact repeated in the reverse order; in the Appendix, as an example, we describe the decryption for RC5.

Thus, the encryption process for RC5, RC6, and many other ciphers can schematically be represented as a chain of "elementary" encryption stages (or rounds)

$$x_1 = \mathrm{Encr}_1(x_0, k_1), \qquad x_2 = \mathrm{Encr}_2(x_1, k_2), \qquad \ldots, \qquad x_r = \mathrm{Encr}_r(x_{r-1}, k_r), \tag{1}$$

where x_0 is an original ℓ-bit word to be encrypted; Encr_i is the encryption operation (function) in the ith stage; k_i is the key used in the ith stage; x_i is an ℓ-bit word, the "output" of the ith stage and the "input" of the $(i+1)$st; finally, x_r is the encryption result.

In the present paper, we describe a new attack on block ciphers of this type, which we call the gradient statistical attack, and analyze, as an example, its applicability to the RC5 cryptanalysis. Obtained experimental results allow us to conclude that the attack is applicable, and for some regimes of the cipher, the complexity of this attack is significantly less than for the direct exhaustive search. Here it should be noted that, though the proposed attack was never described before and is new, analysis of statistical properties of block ciphers has been used in cryptanalysis (see [6, 7]).

The proposed attack is described in Section 2. Section 3 examines its applicability to RC5.

2. DESCRIPTION OF THE STATISTICAL ATTACK

The described method belongs to the class of chosen plaintext attacks (see [1, 3, 5]). When realizing this attack, a cryptanalist may apply any text to the input of the cipher and then analyze the obtained encrypted message. The aim of the attack is to find a (secret) key; the cryptanalist is assumed to know all characteristics of the cipher except for this key. Such attacks are of practical interest, and up-to-date block ciphers are supposed to be resistant to them [1, 3].

We consider ciphers where an encrypted binary word is a block of length ℓ, $\ell > 1$; it is enciphered using a key K, which is a randomly chosen $|K|$-bit word. (Here and in what follows, $|u|$ is the length of u if u is a word and the cardinality of u if u is a set.)

Most of present-day ciphers have an initialization stage, during which an initial key K is transformed into so-called round keys $k_1, k_2, \ldots, k_r$, successively used for encryption in different stages (see (1)). In different ciphers, this procedure goes differently; this depends not only on a particular cipher but also on values of the block and key lengths (ℓ and k) and on the number of rounds r, which for many ciphers are parameters. For instance, for the RC5, the block length can be 32, 64, or 128 bits, the number of rounds can be any integer number, and the key length must be a multiple of 8 and can take any value starting from 8 bits. Note that the values $\ell = 64$, $r = 12$, and $k = 128$ are recommended by the designers and have been extensively studied. Also, schemes where the key length K is the total sum of round key lengths ($|K| = \sum\limits_{i=1}^{r} |k_i|$) are often considered.

Decryption is made in the order reverse to the encryption (1):

$$x_{r-1} = \mathrm{Decr}_r(x_r, k_r), \qquad x_{r-2} = \mathrm{Decr}_{r-1}(x_{r-1}, k_{r-1}), \qquad \ldots, \qquad x_0 = \mathrm{Decr}_1(x_1, k_1), \tag{2}$$

where the same round keys are used, and operations Decr_i are inverse to encryption stages Encr_i.

Let us estimate the complexity of the exhaustive search attack. To make this attack, it suffices to have one encrypted message (binary word) of length not less than the key length. Then we have to attempt decrypting the encrypted message by successively trying all possible keys in some order and comparing the obtained result with the initial (clear) text; coincidence would mean that the unknown key is found. Usually, it is assumed that the key takes any value from the set of all binary words of length $|K|$ with probability $2^{-|K|}$, so the average number of keys searched though equals $\dfrac{2^{|K|}+1}{2}$.

Present-day block ciphers must satisfy a lot of various requirements. One of the requirements can be stated as follows: each encrypted message must "resemble" a realization of a Bernoulli process with generating probability 1/2 for zero and one. (In what follows, we for brief call such sequences random.) In particular, all ciphers that participated in the 21st Century Block Cipher contest held in the USA in 1999–2001 were tested for this requirement (see [8, 9]). We shall not dwell on a logical analysis of this requirement (which, in a sense, cannot be satisfied at all) but give an example clarifying it. To this end, define an ℓ-bit word α_i as a binary notation for the number i, $i = 0, 1, 2, \ldots, 2^\ell - 1$, where, as above, ℓ is the block length of a cipher in question (i.e., α_0 is an ℓ-tuple of binary zeros, α_1 consists of $\ell - 1$ zeros and a unity, α_2 consist of $\ell - 2$ zeros followed by 10, etc.). The requirement on the cipher is that, for any key value, the sequence of ℓ-bit words

$$\mathrm{Encr}(\alpha_0)\,\mathrm{Encr}(\alpha_1)\,\mathrm{Encr}(\alpha_2)\ldots$$

viewed as a binary sequence must be statistically indistinguishable from a random sequence. (Here $\mathrm{Encr}(\alpha_i)$ means the encrypted word α_i.) In particular, this requirement allows one to use a block cipher as a pseudorandom number generator (see [3–5]).

Now we pass to a description of the proposed statistical attack on block ciphers whose encryption and decryption procedures are divided into a series of rounds (1) and (2). We start with quite an informal preliminary consideration. We shall use absolutely inexact terms *more* and *less* random sequences, meaning that one sequence is more random than another if the length required to establish deviation from randomness for sure for the first sequence is larger than that for the second. (Here it is assumed that a certain statistical test is used with the same confidence level. Another definition of a "more" random sequence is that the size of the test statistic for this sequence is less than for a less random one.) Assume that a cipher with an unknown key is successively fed by words $\alpha_0\alpha_1\alpha_2\ldots$. Clearly, this sequence is "highly" nonrandom. The sequence

$$\mathrm{Encr}_1(\alpha_0, k_1)\,\mathrm{Encr}_1(\alpha_1, k_1)\,\mathrm{Encr}_1(\alpha_2, k_1)\ldots$$

obtained after the first encryption round, which we denote by $\beta_0\beta_1\ldots$, is "more" random than the initial one; the sequence

$$\mathrm{Encr}_2(\beta_0, k_2)\,\mathrm{Encr}_2(\beta_1, k_2)\,\mathrm{Encr}_2(\beta_2, k_2)\ldots$$

obtained after the second round is still more random, etc. Finally, the sequence $\omega_0\omega_1\omega_2\ldots$ obtained after the last round is more random than the preceding one. This informal assertion is experimentally verified in data for the RC5 cipher presented below and in numerous researches (see, e.g., [6–9]) for almost all known ciphers of this type; an explanation for this fact is rather obvious: encryption in each round leads to "hashing" and thus increases the "randomness" of encrypted data. Note also an obvious corollary: in decryption of the sequence $\omega_0\omega_1\omega_2\ldots$ according to (2), randomness of the obtained data successively decreases. This, of course, is valid only if "true" round keys are used in the decryption; if, say, in the first decryption round $x_{r-1} = \mathrm{Decr}_r(x_r, k_r)$ (which corresponds to the last encryption round; see (1) and (2)), instead of the true key k_r we use another word k_r^* of the same length, then the effect of the transformation $\mathrm{Decr}_r(x_r, k_r^*)$ will be the same as in encryption, i.e., the output sequence will be more random than the input one. This observation (important for us) in the general form is as follows: if we use a "wrong" key k_j^* in the jth decryption round (instead of the "true" key k_j), then the randomness of the output sequence increases, whereas it decreases if we use the "true" key k_j.

This observation is what the proposed attack is based upon; the attack can now be described schematically in the following way:

1. Problem setting. We are given a cipher whose encryption and decryption is made according to schemes (1) and (2), respectively. All parameters of the cipher except for a key K are assumed to be known. The *goal of the attack* is to find unknown round keys $k_1, k_2, \ldots, k_r$, where, as above, r is the number of rounds (which is equivalent to finding K since this makes it possible to decrypt any message encrypted using this key).

2. Description of the algorithm scheme. In the course of the proposed attack, as the cipher input, we first take a "simple" sequence of m_r ℓ-bit words (say, the above-described $\alpha_0\alpha_1\alpha_2\ldots\alpha_{m_r}$), where m_r is a parameter of the method. Denote the obtained encrypted output sequence by $\omega_0\omega_1\omega_2\ldots\omega_{m_r}$. It is assumed that we use some quantitative measure of randomness, which we denote by $\gamma(w)$, where w is a binary word. (For instance, in the sequel we use as such a measure the statistic applied in the well-known Pearson χ^2 test.)

After that, for all possible values of the rth round key k_r, we successively compute a sequence $\Gamma_r(u)$ defined as

$$\Gamma_r(u) = \mathrm{Decr}_r(\omega_0, u)\,\mathrm{Decr}_r(\omega_1, u)\,\mathrm{Decr}_r(\omega_2, u)\ldots\mathrm{Decr}_r(\omega_{m_r}, u), \tag{3}$$

where $u \in \{0,1\}^{|k_r|}$, and evaluate its "measure" of randomness. Then we find a value of u^* for which the randomness $\gamma(\Gamma_r(u^*))$ of $\Gamma_r(u^*)$ is the smallest among all values of $\gamma(\Gamma_r(u))$, $u \in \{0,1\}^{|k_r|}$, and assume u^* to be the (unknown) key of the rth round: $k_r = u^*$. Let us note here that the number of decryption operations in this stage is proportional to $2^{|k_r|}m_r$.

The we repeat analogous computations to find a key k_{r-1} of the $(r-1)$st round, using as an input the sequence $\Gamma_r(k_r)$ $(= \Gamma_r(u^*))$; see (3). More precisely, we compute the sequence

$$\Gamma_{r-1}(u) = \mathrm{Decr}_{r-1}(\mathrm{Decr}_r(\omega_0, k_r), u)\,\mathrm{Decr}_{r-1}(\mathrm{Decr}_r(\omega_1, k_r), u)\ldots, \tag{4}$$

where now $u \in \{0,1\}^{|k_{r-1}|}$, and evaluate its randomness. We assume that the number of ℓ-bit words in this sequence, which we denote by m_{r-1}, is not larger than m_r (if it is not the case, missing words can be computed, though, as will be seen below, the length of $\Gamma_{r-1}(u)$ [i.e., m_{r-1}] is usually less than the length of $\Gamma_r(u)$ [i.e., m_r] since the first sequence is "less" random than the second). The word u^{**} that minimizes the randomness of $\Gamma_{r-1}(u)$ is taken as the value of the $(r-1)$st round key. In this stage, the number of decryption operations is proportional to $2^{|k_{r-1}|}m_{r-1}$.

Successively repeating the above computations, we find values of the round keys $k_{r-1}, k_{r-2}, k_{r-3}, \ldots, k_1$. The total number of operations required to find all the round keys is proportional to $\sum\limits_{i=1}^{r} 2^{|k_i|}m_i$; in the typical case where all lengths of round keys are the same ($|k_i| = |k|$), the number of operations is proportional to $rm_{\max}2^{|k|}$, whereas for the exhaustive search it is proportional to $2^{|K|}$ (where $m_{\max} = \max\limits_{i=1,\ldots,r} m_i$ and K is the [aggregate] key of the cipher). This difference in the exponents determines the domain of applicability for the proposed attack: if $rm_{\max}$ is less than $2^{|K|-|k|}$, then the number of operations for the proposed method is less than for the exhaustive key search.

3. Modifications, parameters, and variants of the proposed method. We have described the main idea of the method in a "pure" form, and here let us dwell on possible variants of its implementation.

First, the randomness measure $\gamma(\cdot)$ is a parameter of the method; moreover, different measures can be used not only for different ciphers but also for different rounds. As is pointed out above, any statistical test applicable for testing the null hypothesis H_0 that a binary sequence is generated by a Bernoulli source with equal probabilities for zero and one against the alternative hypothesis H_1, the negation of H_0, can be used for this purpose. Here $\gamma(\cdot)$ can be equal to the size of the test statistic.

Second, unlike the variant described above, when searching for the key of the jth round, we may choose not one "true" key but several (say, s) "suspect" values of u, i.e., s words with the smallest measure of randomness $\gamma(\Gamma_j(u))$ (among $u \in \{0,1\}^{|k_j|}$). Furthermore, when finding simple sequences and keys, it is natural to use sequential methods similar to sequential tests in mathematical statistics.

Third, an initial "highly nonrandom" sequence $\alpha_0\alpha_1\alpha_2\ldots\alpha_{m_r}$ can be chosen in various ways. For example, it seems reasonable to choose sequences where neighboring words, α_i and α_{i+1}, not merely contain many identical symbols but differ by one symbol only (such a sequence can be constructed based on Gray codes; their description can be found, e.g., in [10]). Finally, a part of binary symbols in words of a sequence $\alpha_0\alpha_1\alpha_2\ldots\alpha_{m_r}$ can be chosen randomly, the others being set to zero (as in [6, 7]), etc.

The last modification is due to the fact that, for many present-day ciphers, for a large number of rounds even a "highly" nonrandom sequence after encryption is almost indistinguishable from a random one (using known statistical tests with a reasonable computation time). For example, let a cipher have r rounds and assume that, for some "simple" initial sequence $\alpha^0 = \alpha_0^0\alpha_1^0\alpha_2^0\ldots\alpha_m^0$, the sequences

$$\begin{aligned}
\alpha^1 &= \mathrm{Encr}_1(\alpha_0^0, k_1)\,\mathrm{Encr}_1(\alpha_1^0, k_1)\,\mathrm{Encr}_1(\alpha_2^0, k_1)\ldots\mathrm{Encr}_1(\alpha_m^0, k_1),\\
\alpha^2 &= \mathrm{Encr}_2(\alpha_0^1, k_2)\,\mathrm{Encr}_2(\alpha_1^1, k_2)\,\mathrm{Encr}_2(\alpha_2^1, k_2)\ldots\mathrm{Encr}_2(\alpha_m^1, k_2),\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots,\\
\alpha^d &= \mathrm{Encr}_d(\alpha_0^{d-1}, k_d)\,\mathrm{Encr}_d(\alpha_1^{d-1}, k_d)\,\mathrm{Encr}_d(\alpha_2^{d-1}, k_d)\ldots\mathrm{Encr}_d(\alpha_m^{d-1}, k_d)
\end{aligned}$$

are nonrandom for all round keys $k_1, \ldots, k_d$, $d < r$. Then the above-described attack can be modified as follows: for any set of keys $k_{d+1}, \ldots, k_r$ of rounds $d+1, \ldots, r$, we repeat the above-described procedure of finding unknown keys $k_1, \ldots, k_d$ of rounds $1, \ldots, d$. In other words, the keys $k_{d+1}, \ldots, k_r$ are found using exhaustive search, and the keys $k_1, \ldots, k_d$, using the above method. For this combined attack, we need

$$m \cdot 2^{\sum\limits_{j=d+1}^{r} |k_j|} \sum_{j=1}^{d} 2^{|k_j|}$$

operations, which for some parameters may be less than the number of operations required for the exhaustive search of all keys.

3. EXPERIMENTS WITH RC5

We start the description with an experimental analysis of the "measure of randomness" of encrypted messages depending on the number of rounds (more precisely, of half-rounds, as was

Table 1. The number of sequences (out of 100) for which the randomness hypothesis was rejected

Rounds	1	1.5	2	2.5	3
Key number t	2^{18}	2^{18}	2^{18}	2^{20}	2^{20}
1	100	63	64	51	52
2	100	100	100	74	70
3	100	61	61	17	17
4	100	81	78	62	64
5	100	100	100	65	6
6	100	85	86	12	9
7	100	100	100	11	8
8	100	98	99	99	99
9	100	80	79	14	14
10	100	100	100	7	5

mentioned above). We use terms like 3.5 *rounds* instead of, say, *the* 7*th half-round*. Unfortunately, this terminology is commonly used in papers concerning RC5, RC6, and a number of other ciphers.

The first question that we studied experimantally[2] concerned the possibility to distinguish "simple" and evidently nonrandom sequences encrypted by RC5 for various number of (half-)rounds. To do this, we used as an initial "nonrandom" sequence the above-mentioned sequence α_i, $i = 0, 1, \ldots,$ where α_i is the binary notation of length 64 bits for a number i. (Recall that we consider RC5 with block length 64 bits.) In all cases, the sequence was encrypted using this cipher with a given number of half-rounds, and the obtained sequence was used to test the hypothesis H_0 that a binary sequence is generated by a Bernoulli source with equal probabilities of zero and one against the alternative hypothesis H_1, the negation of H_0. In the sequel, to avoid repetitions, we call this problem the "randomness hypothesis."

Clearly, the choice of a statistical test for hypotheses testing plays an important role in the described attack, so let us briefly discuss this question. Presently, there are quite a lot of works devoted to designing and analyzing tests for the randomness hypothesis testing; apparently, this is due to the importance of this problem for cryptography, numerical methods, and other numerous applications. Thus, the US National Institute of Standards and Technology (NIST) has recently analyzed known test for the randomness hypothesis testing and recommended 16 methods for practical application in cryptography (see [11]). In [12] it is shown that the tests described in [13, 14] surpass the methods from [11], which was also justified by our preliminary estimates for RC5. Therefore, for our analysis, we used the "book stack" and "adaptive χ^2" tests from [13, 14], respectively. It was found that the strength of the "book stack" test is on the average higher than that of the adaptive χ^2 test, whereas the latter has a much higher rate and is more convenient to realize on a multiprocessor computer. Therefore, we gave preference to the adaptive χ^2 test; all data given below were obtained for this test.

Table 1 contains data on the randomness hypothesis testing using the adaptive χ^2 test for RC5 with various number of rounds. All computations were made for 10 randomly chosen keys and were repeated 100 times for encrypting the following 100 sequences of words of length t:

$$\alpha_0\alpha_1\ldots\alpha_{t-1}, \qquad \alpha_t\alpha_{t+1}\ldots\alpha_{2t-1}, \qquad \ldots, \qquad \alpha_{99t}\alpha_{99t+1}\ldots\alpha_{100t-1}, \tag{5}$$

where t is the length of one subsequence. The table gives the number of cases where the randomness hypothesis was rejected with confidence level 0.0001. For example, it is seen from the table that the randomness hypothesis was rejected 100 times (out of 100) when using the first (randomly

[2] Computations were made using supercomputers of the Institute of Computational Technologies, Siberian Branch of the RAS, and the Novosibirsk State University.

Table 2. Testing the randomness hypothesis for a larger number of rounds, with confidence level 0.01

Round	Length, t	Number of tests	Number of cases for which the randomness hypothesis was rejected
5	2^{28}	30	30
5.5	2^{29}	22	10
6	2^{31}	6	6
6.5	2^{32}	6	6
7	2^{32}	6	5
7.5	2^{33}	3	3
8	2^{37}	3	2

Table 3. The number of "nonrandom" sequences (out of 100) in decryption with the true key and 10 random half-round keys

2.5 rounds	The length, t, of one sequence (out of 100) is 2^8									
Key Series	True	1	2	3	4	5	6	7	8	9	10
1	54	9	10	10	15	13	13	8	10	19	12
2	69	34	34	35	34	36	33	34	36	39	33
3	87	44	37	36	38	38	37	41	42	42	41

3 rounds	The length, t, of one sequence (out of 100) is 2^{16}									
Key Series	True	1	2	3	4	5	6	7	8	9	10
1	97	81	84	84	81	84	83	84	82	83	83
2	73	35	39	36	32	36	32	33	40	39	42
3	94	0	1	2	0	1	1	0	4	0	1

3.5 rounds	The length, t, of one sequence (out of 100) is 2^{19}									
Key Series	True	1	2	3	4	5	6	7	8	9	10
1	100	28	16	23	9	15	26	18	17	22	22
2	48	9	10	9	11	10	8	9	10	10	11
3	65	20	21	18	20	19	20	19	19	18	17

chosen) key, with sequence length $t = 2^{18}$. Thus, it is seen that encrypted sequences are evidently nonrandom; otherwise, the hypotheses would be rejected in approximately $0.0001 \cdot 100 = 0.1$ cases out of 100.

For a larger number of rounds, we made computations with a smaller number of variants (or repetitions) since this case requires sequences of larger lengths and, accordingly, larger computation time. We again tested the randomness hypothesis H_0 for the same encrypted sequence $\alpha_0\alpha_1\ldots\alpha_{t-1}$ with various (randomly chosen) keys and various number of rounds; results are presented in Table 2. It is seen that an encrypted sequence $\alpha_0\alpha_1\ldots\alpha_{t-1}$ can reliably be distinguished from a random sequence up to round 8.

As is said in the description of the test, the main assumption (unless it is satisfied, the attack is impossible) is as follows: if we use a "wrong" key k^* for decryption in any round (instead of the "true" key k), the randomness of an output sequence increases, whereas it decreases if we use the "true" key k. This assumption was experimentally verified according to the following scheme: for three randomly chosen keys, the above-described 100 sequences (5) were encrypted using RC5 up to the jth half-round. Then one decryption half-round was made using the "true" half-round key and 10 randomly chosen "wrong" keys, and for all the 11 sequences we evaluated the randomness of the data obtained after these transformations. Note that, according to our hypothesis, the difference between the randomness of a sequence decrypted using the "true" key

Table 4. Difference in complexity of sequences decrypted with the "true" half-round key and 5 random keys

Round	Key					
	True	1	2	3	4	5
4	10	4	4	4	3	3
4.5	5	0	0	0	0	0

and of the 10 others, decrypted with random keys, should correspond to the difference in randomness obtained in one extra encryption round. (Indeed, the true key decreases the randomness by one half-round, and a wrong key increases it by one half-round.)

Table 3 contains experimental data for various number of rounds, with confidence level 0.0001. Values of test parameters and the sequence length t were chosen in preliminary experiments with independent data obtained using other random keys.

It turned out that 54 out of 100 sequences decrypted with the true half-round key were recognized as nonrandom (with confidence level 0.0001), whereas only 9 out of 100 sequences "decrypted" with the first wrong key were recognized as nonrandom, 10 out of 100 with the second key, 10 with the third, etc.; so sequences decrypted using the true key are less random than those "decrypted" using wrong keys.

Unfortunately, for a larger number of rounds, computations according to this scheme become practically impossible due to a drastic increase in computation time. (Indeed, using this scheme, we have to make computations for 330 files of the same length: 3 series, 11 half-round keys, and 100 subsequences.) Table 4 shows results for 4 and 4.5 rounds with a sequence (5) of length 2^{24} bits. The sequence was encrypted with a randomly chosen key, and then a decryption half-round was made, once with the "true" half-round key and 5 times with wrong (randomly chosen) keys. These computations were repeated 10 times; all other conditions were the same as in the above-described experiments. It is seen that, for 4 encryption rounds, sequences decrypted with the true key are recognized as nonrandom in 10 cases out of 10, whereas those "decrypted" with a wrong key, only in 4 or 3 cases out of 10. Similarly, for 4.5 rounds, sequences decrypted with the true key were recognized as nonrandom in 5 cases out of 10, and all sequences "decrypted" with wrong keys were recognized as random.

We see that the data given in Tables 1–4 justify the assumptions necessary for the proposed attack to be possible in principle: first, "randomness" of an encrypted sequence grows with the number of half-rounds; second, "randomness" of a sequence "decrypted" with a wrong half-round key is greater than that of a sequence decrypted with the true key.

Thus, the presented experimental results demonstrate that the conditions required for the gradient statistical attack on the RC5 cipher are satisfied. This, in turn, allows us to conclude that this attack on RC5 is possible in principle. Furthermore, the obtained data allow us to suggest that the gradient statistical attack can be applied to other ciphers of the considered type.

APPENDIX: DESCRIPTION OF THE RC5 CIPHER

Algorithm 1 (encryption).

Input: $2w$-bit plaintext $M = (A, B)$; r; key $K = K[0] \dots K[b-1]$.

Output: $2w$-bit ciphertext C.

The encoding uses the operations of addition modulo 2^w ($\boxplus$), XOR ($\oplus$), and cyclic (left) shift ($\hookleftarrow$).

1. Compute $2r + 2$ (half-)round keys $K_0, \dots, K_{2r+1}$ by Algorithm 2, using K and r.

Table 5. The table used for generating half-round keys in RC5

w	16	32	64
P_w	B7E1	B7E15163	B7E15162 8AED2A6B
Q_w	9E37	9E3779B9	9E3779B9 7F4A7C15

```
2. $A \leftarrow A \boxplus K_0$, $B \leftarrow B \boxplus K_0$.
3. Cycle: For $i$ from 0 to $r$ do:
     $A \leftarrow ((A \oplus B) \hookleftarrow B) \boxplus K_{2i}$  (comment: half-round $2i$),
     $B \leftarrow ((B \oplus A) \hookleftarrow A) \boxplus K_{2i+1}$  (comment: half-round $2i+1$).
4. Output $C \leftarrow (A, B)$.
```

Remark. For the decryption, we apply the encryption algorithm using the ciphertext $C = (A, B)$ as follows ($\boxminus$ denotes subtraction modulo 2^w, and $\hookrightarrow$ is the cyclic right shift):

For i from r down to 1 do: $B \leftarrow ((B \boxminus K_{2i+1}) \hookrightarrow B) \oplus A$, $A \leftarrow ((A \boxminus K_{2i}) \hookrightarrow B) \oplus B$.

Finally, we obtain $M \leftarrow (A \boxminus K_0,\ B \boxminus K_1)$.

Algorithm 2 (key initialization).

```
Input: word length $w$; number of rounds $r$; $b$-byte key $K[0] \ldots K[b-1]$.
Output: subkey $K_0, \ldots, K_{2r+1}$ (where $K_i$ are $w$-bit words).
1. Set $u = w/8$ (the number of bytes in a word) and $c = \lceil b/u \rceil$.
2. Next, let $K[j] \leftarrow 0$ for all $b \le j \le c \cdot u - 1$.
     Cycle: For $i$ from 0 to $c-1$ do: $L_i \leftarrow \sum_{j=0}^{u-1} 2^{8j} K[i \cdot u + j]$.
3. $K_0 \leftarrow P_w$; for $i$ from 1 to $2r+1$ do: $K_i \leftarrow K_{i-1} \boxplus Q_w$ (see Table 5).
4. $i \leftarrow 0$, $j \leftarrow 0$, $A \leftarrow 0$, $B \leftarrow 0$, $t \leftarrow \max(c, 2r+2)$. For $s$ from 1 to $3t$ do:
     (a) $K_i \leftarrow (K_i \boxplus A \boxplus B) \hookleftarrow 3$, $A \leftarrow K_i$, $i \leftarrow i + 1 \bmod (2r+2)$.
     (b) $L_i \leftarrow (L_i \boxplus A \boxplus B) \hookleftarrow (A \boxplus B)$, $B \leftarrow L_i$, $j \leftarrow j + 1 \bmod c$.
```

REFERENCES

1. Schneier, B., A Self-Study Course in Block-Cipher Cryptanalysis, *Cryptologia*, 2000, vol. 24, no. 1, pp. 18–34.

2. Biryukov, A., Block Ciphers and Stream Ciphers: The State of the Art, *Cryptology ePrint Archive*, 2004, Report 2004/094. Available at `http://eprint.iacr.org/2004/094/`.

3. Menezes, A.J., van Oorschot, P.C., and Vanstone, S.A., *Handbook of Applied Cryptography*, Boca Raton: CRC Press, 1997.

4. Ryabko, B. and Fionov, A., *Osnovy sovremennoi kriptografii dlya spetsialistov v informatsionnykh tekhnologiyakh*, Moscow: Nauchn. Mir, 2004. Translated under the title *Basics of Contemporary Cryptography for IT Practitioners*, Singapore: World Scientific, 2005.

5. Schneier, B., *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, New York: Wiley, 1996, 2nd ed.

6. Knudsen, R.L. and Meier, W., Correlation in RC6, July 1999. Available at `http://csrc.nist.gov/CryptoToolkit/aes/round2/comments/19990812-lknudsen.pdf`.

7. Shimoyama, T., Takeuchi, K., and Hayakawa, J., Correlation Attack to the Block Cipher RC5 and the Simplified Variants of RC6, in *Proc. 3rd AES Candidate Conf., New York, 2000.* Available at `http://csrc.nist.gov/CryptoToolkit/aes/round2/conf3/papers/36-tshimoyama.pdf`.

8. Nechvatal, J., Barker, E., Bassham, L., Burr, W., Dworkin, M., Foti, J., and Roback, E., Report on the Development of the Advanced Encryption Standart (AES), October 2000. Available at `http://csrc.nist.gov/encryption/aes/round2/r2report.pdf`.

9. Soto, J. and Bassham, L., Randomness Testing of the Advanced Encryption Standard Finalist Candidates, in *Proc. 3rd AES Candidate Conf., New York, 2000*. Available at `http://csrc.nist.gov/encryption/aes/round2/conf3/papers/30-jsoto.pdf`.

10. Knuth, D.E., *The Art of Computer Programming*, vol. 1: *Fundamental Algorithms*, Reading: Addison Wesley, 1973–1981, 2nd ed. Translated under the title *Iskusstvo programmirovaniya na EVM: Osnovnye algoritmy*, Moscow: Mir, 1984.

11. Rukhin, A., Soto, J., Nechvatal, J., Smid, M., Barker, E., Leigh, S., Levenson, M., Vangel, M., Banks, D., Heckert, N., Dray, J., and Vo, S., A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, NIST Special Publication (SP 800-22), 2001. Available at `http://csrc.nist.gov/rng/SP800-22b.pdf`.

12. Ryabko, B.Ya. and Monarev, V.A., Using Information Theory Approach to Randomness Testing, *J. Stat. Plan. Inference*, 2005, vol. 133, no. 1, pp. 95–110.

13. Ryabko, B.Ya., Stognienko, V.S., and Shokin, Yu.I., Adaptive χ^2 Test for Discriminating between Close Hypotheses with a Large Number of Classes and Its Application to Some Cryptography Problems, *Probl. Peredachi Inf.*, 2003, vol. 39, no. 2, pp. 53–62 [*Probl. Inf. Trans.* (Engl. Transl.), 2003, vol. 39, no. 2, pp. 207–215].

14. Ryabko, B.Ya. and Pestunov, A.I., “Book Stack” as a New Statistical Test for Random Numbers, *Probl. Peredachi Inf.*, 2004, vol. 40, no. 1, pp. 73–78 [*Probl. Inf. Trans.* (Engl. Transl.), 2004, vol. 40, no. 1, pp. 66–71].