On the Security of the RC5 Encryption Algorithm

RSA Laboratories Technical Report TR-602
Version 1.0—September 1998

Burton S. Kaliski Jr.
burt@rsa.com

RSA Laboratories East
20 Crosby Drive
Bedford, MA 01730

Yiqun Lisa Yin
yiqun@rsa.com

RSA Laboratories West
2955 Campus Drive
San Mateo, CA 94403

Part number: 003-903075-100-001-000

Contents

Part I
Security of RC5

1 Introduction

The RC5 encryption algorithm was designed by Professor Ronald Rivest of MIT and first published in December 1994 [17]. Since its publication, RC5 has attracted the attention of many researchers in the cryptographic community in efforts to accurately assess the security offered. In this report, we will focus our discussions on the security of RC5 against differential and linear cryptanalysis, but we will also give a brief summary of other known cryptanalytic results on RC5.

The analysis of a cryptographic algorithm is of course essential to its acceptance and use. We observe that the lengthy analysis of the Data Encryption Standard [16] prior to publication, though not public, resulted in an algorithm that has resisted attack for many years. Our hope is that this report will provide a foundation for similarly robust analysis of RC5 by the cryptographic community. In this way any weaknesses can be found early, and so that if RC5 or its enhancements (e.g., RC6 [18]) survive the process it will be suitable as one of the potential successors to DES. We welcome critical comments on this report, and additional approaches to analyzing RC5.

RSA Laboratories' analysis of RC5 is still in progress, and this report will be periodically updated to reflect any additional findings.

2 Description and Features of RC5

RC5 is a parameterized algorithm, and a particular RC5 algorithm is designated as RC5-$w/r/b$. We summarize these parameters below:

- w The *word size*, in bits. The standard value is 32 bits; allowable values are 16, 32, and 64. RC5 encrypts two-word blocks so that the plaintext and ciphertext blocks are each $2w$ bits long.
- r The number of rounds. Allowable values are 0, 1, ..., 255.
- b The number of bytes in the secret key K. Allowable values of b are 0, 1, ..., 255.

RC5 consists of three components: a *key expansion* algorithm, an *encryption* algorithm, and a *decryption* algorithm. These algorithms use the

following three primitive operations (and their inverses).

1. Addition of words modulo 2^w, denoted by "+".
2. Bit-wise exclusive-OR of words, denoted by $\oplus$.
3. Rotation: the rotation of x to the left by y bits is denoted by $x \lll y$. Note that only the $\log_2(w)$ low-order bits of y affect this rotation.

2.1 Key expansion

The key-expansion algorithm expands the user's key K to fill the expanded key table S, so that S resembles an array of $t = 2(r+1)$ random binary words determined by K. It uses two "magic constants" and consists of three simple algorithmic parts.

The two word-size magic constants P_w and Q_w are defined for arbitrary w as follows:

$$\begin{aligned} P_w &= \text{Odd}((e-2)2^w) \\ Q_w &= \text{Odd}((\phi-1)2^w) \end{aligned}$$

where

$$\begin{aligned} e &= 2.718281828459... \text{ (base of natural logarithms)} \\ \phi &= 1.618033988749... \text{ (golden ratio)} , \end{aligned}$$

and where $\text{Odd}(x)$ is the odd integer nearest to x (rounded up if x is an even integer, although this won't happen here).

The first algorithmic step of key expansion is to copy the secret key $K[0, ..., b-1]$ into an array $L[0, ..., c-1]$ of $c = \lceil b/u \rceil$ words, where $u = w/8$ is the number of bytes/word. This operation is done in a natural manner, using u consecutive key bytes of K to fill up each successive word in L, low-order byte to high-order byte. Any unfilled byte positions of L are zeroed. In the case that $b = c = 0$, we reset c to 1 and $L[0]$ to zero.

The second algorithmic step of key expansion is to initialize array S to a particular fixed (key-independent) pseudo-random bit pattern, using an arithmetic progression modulo 2^w determined by the "magic constants" P_w and Q_w. Since Q_w is odd, the arithmetic progression has period 2^w.

$S[0] = P_w$;
for $i = 1$ **to** $t-1$ **do**
    $S[i] = S[i-1] + Q_w$;

The third algorithmic step of key expansion is to mix in the user's secret key in three passes over the arrays S and L. More precisely, due to the potentially different sizes of S and L, the larger array will be processed three times, and the other array may be handled more times.

$i = j = 0;$
$A = B = 0;$
do $3 * \max(t, c)$ **times:**
$\quad A = S[i] = (S[i] + A + B) \lll 3;$
$\quad B = L[j] = (L[j] + A + B) \lll (A + B);$
$\quad i = (i + 1) \bmod(t);$
$\quad j = (j + 1) \bmod(c);$

Note that the key-expansion function has a certain amount of "one-wayness": it is not so easy to determine K from S.

2.2 Encryption and decryption

The description of the encryption algorithm is given in the pseudo-code below. We assume that the input block is given in two w-bit registers A and B, and that the output is also placed in the registers A and B.

$A = A + S[0]$
$B = B + S[1]$
for $i = 1$ **to** r **do**
$\quad A = ((A \oplus B) \lll B) + S[2i]$
$\quad B = ((B \oplus A) \lll A) + S[2i + 1]$

The decryption routine is easily derived from the encryption routine.

2.3 Features of RC5

RC5 is a fast block cipher designed to be suitable for both software and hardware implementation. It is a parameterized algorithm, with a variable block size, a variable number of rounds, and a variable-length secret key. This provides the opportunity for great flexibility in both the performance characteristics and the level of security.

One significant feature of the design of RC5 is its simplicity; encryption is based on only three operations: addition, exclusive-or, and rotation. Thus, it makes RC5 both easy to implement, and very importantly, more amenable to

analysis than many other block ciphers. The connection between simplicity of design and simplicity of analysis, was indeed one of Rivest's goals.

Another distinguished feature of RC5 is the heavy use of data-dependent rotations in encryption. As we will see in this report, this feature is very useful in preventing differential and linear cryptanalysis.

3 Techniques for Analyzing Block Ciphers

Several techniques have been developed for analyzing the security of block ciphers. In this section, we give a brief review of the techniques that will be used in this report, including exhaustive search, statistical tests, differential cryptanalysis, and linear cryptanalysis. The reader can find detailed discussions about these different techniques in [19].

The most basic attack that can always be mounted on a block cipher is that of *exhaustive search*. (If this is also the best attack available, then the designer of the cipher has done a good job!) In such an attack, an adversary obtains a plaintext and its corresponding ciphertext under the secret key and simply tests each of the possible candidates for the key until a match is found. If the key has n bits, then there are 2^n possible keys to test, and hence the amount of work for exhaustive search is closely related to the key size. When key size is larger then the block size, multiple pairs of plaintext/ciphertext may be needed in an exhaustive search attack.

Statistical tests can be used for analyzing the statistical behavior of block ciphers. A strong block cipher should behave like a random permutation of the plaintext for a random key so that it is impossible to get information about the key or plaintexts from ciphertexts except by exhaustive search. Commonly used statistical tests include randomness tests on ciphertext, correlation tests between plaintext, key, and ciphertext, etc. We want to remark that good statistical behaviors are only a necessary condition for the security of block ciphers, and that block ciphers that pass such statistical tests may well still remain catastrophically weak.

Differential cryptanalysis [2], pioneered by Biham and Shamir, has had a quite revolutionary effect on the design and analysis of block ciphers. The basic idea in this technique is the following: Two plaintexts are chosen with a certain "difference" P' between them. Typically, the "difference" is measured by exclusive-or $\oplus$, but for some ciphers an alternative measure can be more useful. These two plaintexts are enciphered to give two ciphertexts such that their difference C' has a specific value with better than average probability. Such a pair (P', C') is called a *characteristic*. Depending on the

cipher and the analysis, the behavior of these characteristics can be useful in deriving certain bits of the key.

Linear cryptanalysis [14], introduced by Matsui, is another theoretical breakthrough in block cipher cryptanalysis. The basic idea of this technique is to find relations among certain bits of plaintext, ciphertext, and key that hold with a probability $p \neq 1/2$ (i.e., bias $= |p - 1/2| > 0$). Such a relation is called a *linear approximation*. Just as in differential cryptanalysis, we seek to exploit such non ideal behavior and it may be possible to identify linear approximations that can be used to obtain information about the key.

4 Summary of Known Cryptanalytic Attacks on RC5

The first cryptanalytic results on RC5 were given by Kaliski and Yin [7] at Crypto'95. By analyzing the basic structure of the encryption routine as well as the properties of data-dependent rotations, they were able to construct differential characteristics and linear approximations of RC5 that are useful for mounting differential and linear attacks. Their results also show that the use of data-dependent rotations and the incompatability between the different arithmetic operations used in encryption help prevent both attacks.

Subsequent results on RC5 are mostly in the area of differential cryptanalysis. At Crypto'96, Knudsen and Meier [9] presented improvements over Kaliski and Yin's differential attack by carefully analyzing the relations among input, output, and subkeys in the first two rounds. Even though the characteristics used in their attack are essentially the same as in [7], they were able to improve the plaintext requirement by exploiting the characteristics in a more sophisticated way at the beginning and the end of the r rounds. They also showed the existence of a small fraction of "differentially weak keys" for RC5 with respect to which their attack can be further enhanced.

Kaliski and Yin [8] further studied how the data-dependent rotation in a single round can spread a small difference in input to a big difference in output. Such a property of data-dependent rotations makes standard differential cryptanalysis infeasible for RC5 with enough rounds.

At Eurocrypt'98, Biryukov and Kushilevitz [3] presented nice improvements over Knudsen and Meier's differential attacks on RC5. They studied more complex differentials than in previous works and define a more general notion of "good pairs" with respect to data-dependent rotations. In particular, all plaintext/ciphertext pairs that escape differences in rotation

amounts can be used, not just pairs that follow specific patterns. Biryukov and Kushilevitz also proposed more efficient methods for finding good pairs. They estimated that RC5 with 12 rounds and 64-bit block size can be attacked using about 2^{44} plaintexts.

Unlike the situation with differential cryptanalysis in which we have seen big improvements over the first attack, RC5 has appeared to be extremely resistant to linear cryptanalysis. Moriai, Aoki, and Ohta [15] investigated the strength of RC5 against linear cryptanalysis by focusing on the bias of linear approximations for *fixed* keys, rather than the average bias (see §9.1) over *all* keys. They also considered a mini-version of RC5 with much reduced word size and computed the percentage of keys that yield ciphers less resistant to linear cryptanalysis than the average case analysis might suggest. Selcuk [21] implemented the first linear attack [7] and showed that the success rate of the attack is much less than the early theoretical estimates due to some hidden assumptions.

As of this writing, the differential attack on RC5 described in [3] remains as the best published result. A summary of the data requirements[1] for this attack with a varying number of rounds is provided in Table 1 for RC5 with a 64-bit block size. The second row in the table has been derived from the first row using the simple fact [2] that a differential attack with m chosen plaintexts can be converted into one with approximately $2^w(2m)^{1/2}$ known plaintexts where the block size is $2w$.

Number of rounds	4	6	8	10	12	14	16	18
Differential attack (chosen plaintext)	2^7	2^{16}	2^{28}	2^{36}	2^{44}	2^{52}	2^{61}	$>$
Differential attack (known plaintext)	2^{36}	2^{41}	2^{47}	2^{51}	2^{55}	2^{59}	2^{63}	$>$

Table 1: Plaintext requirements for the currently best-known attack on RC5 (64-bit block size).

Kocher [10] developed what are called timing attacks that are generally applicable to many cryptosystems. In such an attack, an opponent tries to

[1] While most of the data requirements are impractical anyway, we use "$>$" to denote when the attack is infeasible even at a theoretical level. This is when the plaintext requirements are greater than 2^{2w}, which is the maximum number of possible $2w$-bit plaintexts.

obtain information about the secret or private key by recording and analyzing the time used for cryptographic operations that involve the key. Kocher observed that RC5 may be subject to timing attacks if RC5 is implemented on platforms for which the time for computing a single rotation is proportional to the rotation amount. However, RC5 can easily be implemented in such a way as to be invulnerable to timing attacks. Many modern processors have constant-time rotation, addition, and exclusive-or instructions. Other processors may have a rotation or shift time that depends linearly with the amount of rotation, but in this case it is usually easy to arrange the work so that the total compute time is data-independent, for example, by computing a rotate of t bits using a left-shift of t bits and a right-shift of $w - t$ bits. In either case, the RC5 encrypt/decrypt time is data-independent, causing any potential timing attacks to fail.

With regards to the less sophisticated brute-force attack of trying each key in turn, the security of RC5 is obviously dependent on the length of the encryption key that is used (as is the case with all ciphers). RC5 has the attractive feature that the length of the key can be varied (unlike with DES for instance) and so the level of security against these attacks can be tuned to suit the application. With the launch of the *RSA Data Security Secret-Key Challenge* [20], it is hoped that the resistance of ciphers to exhaustive key search attacks can be more accurately gauged in the future. To help in this assessment, various texts encrypted with RC5 with different length keys have been posted as a challenge to the community. Some of these challenges, such as RC5 with a 40-bit, 48-bit and 56-bit key were solved within a number of months of the announcement of the Challenge [20], as was expected. It is anticipated that some of the longer key lengths will remain an unsolved challenge for some considerable time to come.

5 The Current Status of RC5

The results to date on the cryptanalysis of RC5 have been very encouraging. We observe that RC5 with 12 rounds and 64-bit block size give roughly the same security as DES against analytical attacks – 2^{44} chosen plaintext pairs for RC5 as opposed to 2^{43} known plaintexts for DES. The extra speed of RC5 allows one to use extra rounds, thereby providing an additional margin of safety. Based on the known results, we conclude that RC5 with 16 rounds and 64-bit block size can provide good security against existing analytical attacks.

With the cipher receiving considerable attention from cryptanalysts world-

wide, a picture of the security offered by RC5 has been quick to develop. Acceptance of the cipher is growing, and RC5 has been discussed for inclusion in various standards efforts and has been published by the IETF in RFC2040 [1]. Three years on, it seems that the RC5 block cipher offers a computationally inexpensive way of providing secure encryption.

Part II
Detailed Analysis of RC5

6 Notation

In Rivest's description of RC5 [17], a round consists of two equations, and in each equation, either A or B is modified while the other remains unchanged. We will refer to each equation as a *half-round*. So one half-round of RC5 is similar to a full round in DES [16]. For ease of discussions, we adopt the common notation for Feistel ciphers[2] and rewrite RC5 as follows.

$L_1 = L_0 + S_0$
$R_1 = R_0 + S_1$
for $i = 2$ **to** n **do**
    $L_i = R_{i-1}$
    $R_i = ((L_{i-1} \oplus R_{i-1}) \lll R_{i-1}) + S_i$

We will use the above description of RC5 throughout the report. We will refer to the two equations which involve (L_{i-1}, R_{i-1}) and (L_i, R_i) as the i^{th} half-round of RC5. Hence, the two initial equations ($L_1 = L_0 + S_0$ and $R_1 = R_0 + S_1$) together are considered as the *first* half-round, and RC5 contains $n = 2r + 1$ half-rounds in total. The input block (plaintext) is (L_0, R_0) and the output block (ciphertext) is (L_n, R_n). For ease of notation, we will change $S[i]$ to S_i.

Some additional notation is as follows. For a binary vector x of length w, we label the bit positions from the most significant bit to the least significant bit as $w-1, \ldots, 1, 0$. We use $x[s]$ to denote the s^{th} bit of x and $x[s..t]$ $(s \geq t)$ to denote the s^{th} through t^{th} bits of x. Finally, we use $\lg(w)$ to denote $\log_2(w)$. Note that $x \bmod w = x[\lg(w) - 1..0]$ are the bits of x that are used to determine a rotation by x.

[2]Strictly speaking, RC5 is not a Feistel cipher, since the round function of a Feistel cipher has the general form of $R_i = L_{i-1} \oplus f(R_{i-1}, S_i)$.

7 A General Idea for Attacking RC5

In this section, we describe a general idea for attacking RC5 by analyzing the structure of the RC5 encryption routine. The idea is used in both our differential and our linear cryptanalysis. Note that to attack RC5, one can try to find either the original secret key or the expanded key table S. If the latter approach is used, then the attack is independent of the length of the secret key. In this report, we will focus on the latter approach.

The general idea is to reduce the problem of computing the entire expanded key table S to the problem of computing $L_{n-1}[b]$ for some $0 \leq b \leq w-1$. (Note that $L_{n-1}[b]$ is a bit in the next-to-last half-round and is not visible from the ciphertext.) At a high level, the reduction is accomplished in the following two steps.

1. Reduce the problem of computing S to the problem of computing the last subkey S_n. This is based on the *iterative* structure of the encryption routine.

2. Reduce the problem of computing S_n to the problem of computing $L_{n-1}[b]$. This is based on the structure of the last half-round.

In what follows, we focus on the last half-round and explain in more detail how the reduction works in step 2. Consider the two equations in the last half-round:

$$\begin{aligned} L_n &= R_{n-1}, \\ R_n &= ((L_{n-1} \oplus R_{n-1}) \lll R_{n-1}) + S_n. \end{aligned}$$

There are four variables in the second equation, and two of them, R_n and $R_{n-1}(= L_n)$, are known from the ciphertext. Therefore, if we can obtain information about L_{n-1}, it will immediately give us information about the subkey S_n. To make such a relation concrete, we establish an equation that relates certain bits of the four variables for each *fixed* rotation amount $R_{n-1} \bmod w$.

We first consider a special case where $(b + R_{n-1}) \bmod w = 0$. In this case, the bit $L_{n-1}[b] \oplus R_{n-1}[b]$ moves to bit position 0 after the rotation. We thus have

$$R_n[0] = (L_{n-1}[b] \oplus R_{n-1}[b]) \oplus S_n[0]. \tag{1}$$

Since $R_n[0]$ and $R_{n-1}[b]$ are known, if we can compute $L_{n-1}[b]$, then we can obtain $S_n[0]$, the least significant bit of subkey S_n.

The general case where $(b + R_{n-1}) \bmod w = s$ is a little more involved since there is a carry effect due to the addition of S_n when $s \neq 0$. Let

$$Y = (L_{n-1} \oplus R_{n-1}) \lll R_{n-1},$$

and so

$$R_n = Y + S_n.$$

Let

$$carry(s) = \text{carry out from } Y[s-1..0] + S_n[s-1..0].$$

Then we have that

$$\begin{aligned} R_n[s] &= Y[s] \oplus S_n[s] \oplus carry(s) \\ &= (L_{n-1}[b] \oplus R_{n-1}[b]) \oplus S_n[s] \oplus carry(s). \end{aligned} \tag{2}$$

If $S_n[s-1..0]$ is known, then given a ciphertext (L_n, R_n), we can compute the carry out $carry(s)$ by comparing $S_n[s-1..0]$ with $R_n[s-1..0]$. Once we obtain both $carry(s)$ and $L_{n-1}[b]$, we can compute $S_n[s]$.

We are now in a position to give the full details of the reduction in step 2. Let $\mathcal{B}$ denote an algorithm which computes $L_{n-1}[b]$ given a plaintext/ciphertext pair. Figure 1 contains pseudocode for computing S_n using algorithm $\mathcal{B}$.

```
for s = 0 to w − 1
  select a plaintext/ciphertext pair (L_0, R_0)/(L_n, R_n)
    such that (b + R_{n−1}) mod w = s
  compute L_{n−1}[b] using algorithm B
  if s = 0, then carry(0) = 0
  if s ≥ 1
    if S_n[s − 1..0] ≤ R_n[s − 1..0]
    then carry(s) = 0
    else carry(s) = 1
  S_n[s] = L_{n−1}[b] ⊕ R_{n−1}[b] ⊕ carry(s)
```

Figure 1: Pseudocode for computing the last subkey S_n.

Assuming that RC5 is a pseudorandom function, the rotation amount $s = R_{n-1} \bmod w = L_n \bmod w$ is random for a randomly chosen plaintext.

Thus, when enough random plaintexts are gathered, all possible values of s will occur, and hence all bits of S_n can be recovered.

From the above discussions, we see that an algorithm that can compute $L_{n-1}[b]$ is very useful for recovering S_n. By the reduction in step 1, the same algorithm can also be used to recover other subkeys. More specifically, when we try to recover subkey S_i ($i < n$), we can "unwrap" $n-i$ half-rounds using subkeys $S_{i+1}, \ldots, S_n$ (which are already known) to obtain the outputs from the i^{th} half-round (the corresponding "ciphertexts" of S_i). Then we can compute $L_{i-1}[b]$ and S_i in a similar fashion (See Figure 1).

We remark that there may be other algorithms for computing the bits of L_{n-1}. If so, such algorithms could be extended to an attack against RC5 using the basic idea that we have described in this section. Furthermore, there may be other attacks than differential and linear cryptanalysis to which the techniques described in this section may apply. At this time, however, no alternative effective techniques are known to exit.

8 RC5 and Differential Cryptanalysis

In this section, we will study the security of RC5 against differential cryptanalysis. We will present the details of the first differential attack [7] on RC5. The techniques used in this attack is quite illustrative: they show how to form characteristics for RC5 and how to use certain special characteristics at the end of the r rounds to effectively compute the subkeys. We will also summarize the key ideas in the two subsequent improved differential attacks on RC5 [9, 3].

Later in the section, we will discuss the role of data-dependent rotations in helping prevent differential attacks. Finally, we analyze what are called Markov properties of RC5. Such properties are interesting since they potentially allow one to make additional claims on the resistance of a cipher to differential style attacks.

8.1 The first differential attack on RC5

8.1.1 Characteristics for a half-round of RC5

Roughly speaking, a characteristic for a half-round consists of an input difference and output difference together with the associated probability. Following the notation in [2], we denote such a characteristic by $\Omega = (\Omega_P, \Omega_T)$, where

$$\begin{aligned}\Omega_P &= (L'_{i-1}, R'_{i-1}) = (L_{i-1} \oplus L^*_{i-1}, R_{i-1} \oplus R^*_{i-1}),\\ \Omega_T &= (L'_i, R'_i) = (L_i \oplus L^*_i, R_i \oplus R^*_i).\end{aligned}$$

Intuitively, if a pair of inputs to a half-round have different rotation amounts, then the pair of outputs from the half-round will differ in many different ways (see §8.3 for an analytical justification). Consequently, we will focus on characteristics for which the pair of inputs have the same rotation amounts. Let e_s denote the w-bit binary vector which is 1 in bit s and 0 everywhere else. For most of the characteristics that we present below, each half of Ω_P and Ω_T is either zero or e_s for $s \geq \lg(w)$, implying that the rotation amounts will be the same.

We will calculate the probability associated with a half-round characteristic by averaging over both the pair of inputs and subkey S_i. This is for the reason of simplicity. There may be keys for which the probability is higher and others for which it is lower. However, assuming the key expansion of RC5 is good, subkeys will be essentially independent of one another, and hence the overall probability of a characteristic for n half-rounds will be close to what we would expect for nearly all keys. Implementation results also confirm that this appears to be reasonable.

Table 2 lists five half-round characteristics that will used in the differential attack. When analyzing these probabilities, we use the fact that for random inputs x and y with $x \oplus y = e_s$ and random key S_i, the probability that $(x + S_i) \oplus (y + S_i) = e_s$ is at least $1/2$.

Ω	Ω_P	Ω_T	conditions	probability
Ω^1	$(0, e_s)$	(e_s, e_s)	$s \geq \lg(w)$	$p \geq \frac{1}{w} \cdot \frac{1}{2}$
Ω^2	(e_s, e_s)	$(e_s, 0)$	$s \geq \lg(w)$	$p = 1$
Ω^3	$(e_s, 0)$	$(0, e_t)$	$s, t \geq \lg(w)$	$p \geq \frac{1}{w} \cdot \frac{1}{2}$
Ω^4	$(0, e_s)$	(e_s, e_t)	$s, t \geq \lg(w), t \neq s$	$p \geq \frac{1}{w} \cdot \frac{1}{2}$
Ω^5	(e_s, e_t)	$(e_t, e_u \oplus e_v)$	$s, t \geq \lg(w), t \neq s, u > v$ $t - s = \pm(u - v) \bmod w$	$p \geq \frac{1}{w} \cdot \frac{1}{2} \cdot \frac{1}{2}$

Table 2: Useful characteristics for a single half-round.

For characteristics Ω^3, Ω^4, and Ω^5, there are many possible output differences Ω_T for each input difference Ω_P. In particular, for each choice of Ω_P, there are $(w - \lg(w))$ choices of parameter t for Ω^3, $(w - \lg(w) - 1)$ choices of parameter t for Ω^4, and w choices of parameters (u, v) for Ω^5.

For the first half-round, there are three characteristics that hold with probability 1:

$\Omega^{1'} : \Omega_P = \Omega_T = (0, e_{w-1})$, which may be joined with Ω^1,

$\Omega^{2'} : \Omega_P = \Omega_T = (e_{w-1}, e_{w-1})$, which may be joined with Ω^2, and

$\Omega^{3'} : \Omega_P = \Omega_T = (e_{w-1}, 0)$ which may be joined with Ω^3.

These characteristics are particularly useful.

8.1.2 Characteristics of RC5

In this section, we show how to join the half-round characteristics described in §8.1.1 to form characteristics for RC5 in its entirety.

We first note that two characteristics can be joined together if the output difference Ω_T of the first one and the input difference Ω_P of the second one are the same. For example, Ω^3 with parameters (s_1, t_1) can be joined to Ω^1 with parameter s_2 if $t_1 = s_2$. Therefore, the possible ways to join the five characteristics in Table 2 are Ω^1-Ω^2, Ω^2-Ω^3, Ω^3-Ω^1, Ω^3-Ω^4, and Ω^4-Ω^5. (Ω^1 may be viewed as a special case of Ω^4 in which $s = t$. It is useful to distinguish between them since Ω^1 cannot be joined with Ω^5.)

Two particular ways of joining the half-round characteristics will be especially useful: The first one is $\bar{\Omega} = \Omega^1$-Ω^2-Ω^3, a characteristic for three half-rounds that can be repeatedly joined with itself. The second one is Ω^4-Ω^5, giving a characteristic for two half-rounds that can be used to compute $L_{n-1} \bmod w$. (More details including generalizations of Ω^4-Ω^5 are given in §8.1.3.)

Based on the earlier discussions, we can now construct characteristics for n half-rounds of RC5, which we will denote by Ω_n. Characteristic Ω_n consists of a sequence of half-round characteristics. Since there are many possible values for the parameters of some of the half-round characteristics, there are many possible paths (corresponding to many intermediate differences (L'_i, R'_i) for $1 \le i \le n-1$) from P' to C' for Ω_n, all of which have the same probability p. If we let N denote the total number of possible paths for Ω_n, then we define the probability associated with Ω_n as $p^{\Omega_n} = Np$.

For different values of n, Table 3 lists the plaintext difference P', the sequence of half-round characteristics in Ω_n, and the probability[3] given by

[3](1) The factor $\frac{1}{4}$ in Ω^5 in Table 3 can be mostly eliminated by taking the carry effect

p^{Ω_n}.

n	P'	Ω_n	p^{Ω_n}
$3m$	$(0, e_{w-1})$	$\Omega^{1'}$-$\bar{\Omega}$- $\cdots$-$\bar{\Omega}$-Ω^4-Ω^5	$\frac{w-\lg(w)-1}{w}\left(\frac{w-\lg(w)}{(2w)^2}\right)^{m-1}$
$3m+1$	$(e_{w-1}, 0)$	$\Omega^{3'}$-Ω^3-$\bar{\Omega}$- $\cdots$-$\bar{\Omega}$-Ω^4-Ω^5	$\frac{w-\lg(w)-1}{1}\left(\frac{w-\lg(w)}{(2w)^2}\right)^{m}$
$3m+2$	(e_{w-1}, e_{w-1})	$\Omega^{2'}$-Ω^2-Ω^3-$\bar{\Omega}$- $\cdots$-$\bar{\Omega}$-Ω^4-Ω^5	$\frac{w-\lg(w)-1}{1}\left(\frac{w-\lg(w)}{(2w)^2}\right)^{m}$

Table 3: Useful characteristics for n half-rounds and their associated probability.

A *right pair* with respect to Ω_n consists of two plaintexts P, P^* and their ciphertexts C, C^* such that for all $0 \le i \le n$, the corresponding difference (L'_i, R'_i) has a form specified by one of the sequences of the half-round characteristics for Ω_n. For $i \le n-1$, a characteristic Ω_i, its associated probability p^{Ω_i}, and a right pair with respect to Ω_i can be defined in a similar way.

Note that the type of the characteristics used in the differential attack on RC5 is quite different from the characteristics used in attacks on other block ciphers, e.g. DES. In particular, for a given plaintext difference P' and ciphertext difference C', there are many possible paths (intermediate differences) from P' to C', each occuring with the same probability. This differential effect help boost the probability of getting a right pair.

8.1.3 Using right pairs to compute the subkeys

Here we first show how to compute the last subkey S_n using a right pair with respect to the characteristic Ω_n. Then we analyze the number of right pairs needed to recover every bit of S_n. For $i < n$, subkey S_i can be obtained similarly using right pairs with respect to Ω_i, following the reduction method we outlined in §7.

Let Ω^4 and Ω^5 be the characteristics for the $(n-1)^{th}$ and n^{th} half-rounds, respectively. Let s, t, u, v be the parameters for Ω^5 so that s, t are the parameters for Ω^4. By considering the $(n-1)^{th}$ half-round, we can obtain the following formula:

$$L_{n-1} \bmod w = R_{n-2} \bmod w = (t-s) \bmod w.$$

into account when analyzing output differences. Hence the factor does not appear in p^{Ω_n} in Table 4. (2) When $n = 3m$, the probability associated with the first occurrence of the half-round characteristic Ω^1 is $\frac{1}{w}$ instead of $\frac{1}{2w}$ since the parameter $s = w-1$.

Given the ciphertext difference (L'_n, R'_n), the values of t, u, v are easily obtained from the form of Ω^5. So we need only compute s in order to get $L_{n-1} \bmod w$. In the n^{th} half-round, the rotation amount $L_n \bmod w$ $(= R_{n-1} \bmod w)$ is equal to either $(u - t) \bmod w$ or $(v - t) \bmod w$. Since u, v, t, and L_n are known, it is obvious which case holds. In the first case $s = (v - L_n) \bmod w$ and in the second case $s = (u - L_n) \bmod w$, and the value of $L_{n-1} \bmod w$ follows.

The key idea in the above analysis is the following:

- A certain *pattern* of the two differences L'_{n-1}, R'_{n-1} can reveal the rotation amount $L_{n-1} \bmod w$.
- The pattern can be derived from the ciphertexts.

There may be many possible characteristics for the last two half-rounds that satisfy the above two conditions. The characteristic (Ω^4, Ω^5) is just one of them, and it is one with small Hamming weights (the number of 1's in a binary vector) in the ciphertext difference. (See §8.2 for discussions on other possible characteristics.)

Below, we analyze the number of right pairs needed to recover every bit of S_n, and we denote this number by T. We have seen that each right pair allows us to compute $L_{n-1}[\lg w - 1..0]$. Based on the discussions in §7, we can therefore compute $\lg w$ consecutive bits of S_n. The bit positions depend on the rotation amount $L_n \bmod w$, which can be assumed to be random for a random right pair. Hence, the probability that there exists a bit $S_n[s]$ which it cannot be computed from any of the T random pairs is at most

$$w[(w - \lg w)/w)]^T.$$

If we set $T = 2w$, the above probability is less than 1% for $w = 16, 32, 64$.

8.1.4 Analyzing plaintext requirements

In this section, we will analyze the plaintext requirements for implementing a differential attack on RC5 using the characteristics derived in the previous sections. We will address the issue of noise in the analysis.

We defined the notion of a right pair in §8.1.2, and here we introduce the notion of a good pair. Formally, a *good pair* with respect to characteristic Ω_n consists of two plaintexts P, P^* and their ciphertexts C, C^* such that the input and output difference (P', C') satisfies the condition of a right pair with respect to the same characteristic. When implementing a differential attack in practice, we can only observe good pairs, as opposed to right pairs.

A good pair is not necessarily a right pair with respect to Ω_n due to certain *noise*—the sequence of intermediate differences follows a path different from the one specified by Ω_n. We consider two types of noise:

1. Random noise. For a random pair of plaintexts (that may not be a good pair), the probability that the pair of ciphertexts have the difference $C' = \Omega_T$ is

$$p^{rand} = \frac{(w - \lg w) \cdot w(w-1)/2}{2^{2w}}.$$

   This noise is negligible when compared to p^{Ω_n} (the probability of a right pair) if $n \leq 23$ (i.e. $r \leq 11$). When $r \geq 12$, the noise becomes dominating.

2. Special noise. For a random good pair (having a fixed plaintext difference $P' = \Omega_P$, there is a non-negligible probability that it is not a right pair due to the special difference P'. To see how this can happen, we recall the characteristics for the last five half-rounds in a right pair. The number of non-zero bits in (L'_i, R'_i) for $i = n-4, \ldots, n$ are the following:

$$(1,1), (1,0), (0,1), (1,1), (1,2).$$

   A pair of plaintexts with difference P' may follow the correct intermediate differences until the $(n-5)^{th}$ half-round and then have the following number of non-zero bits in the last five half-rounds:

$$(1,1), (1,2), (2,1), (1,1), (1,2).$$

   This happens for a fraction of the good pairs, and yields good pairs that are not right pairs. In general, the intermediate differences can be more complicated and happen with a lower probability. Implementation results show that the fraction of good pairs that are not right pairs is no more than 10% for $w = 32$.

Bringing all this information together, we now compute the number of good pairs needed for an attack with a high success rate. When $n \leq 23$, p^{rand} can be ignored. If we generate $2w$ good pairs, then on average there are $2w \lg(w)/w = 2 \lg(w)$ good pairs that are useful for predicting the value of each bit $S_n[s]$. With high probability, more than half of the good pairs are right pairs, so a majority vote will yield the correct value of $S_n[s]$. Therefore, $2w$ good pairs are enough for $n \leq 23$.

As n gets larger, p^{Ω_n} will eventually become smaller than p^{rand} as noted above, and so more good pairs will be needed in the attack. For RC5-32, $n = 24$ is the starting point at which p^{Ω_n} becomes smaller than p^{rand}. In this case, $8w$ good pairs are needed to guarantee a high success rate.

The expected number of plaintext pairs required for computing the last subkey S_n is the product of (1) the number of good pairs needed and (2) the expected number of plaintext pairs to get a single good pair ($\leq \frac{1}{p^{\Omega_n}}$) (see Table 3). For RC5-32/r/b (64-bit block size), the number of chosen plaintext pairs are listed for increasing r ($1 \leq r \leq 12$) in Table 4.

r	plaintexts	r	plaintexts	r	plaintexts
1	2^8	5	2^{26}	9	2^{46}
2	2^{11}	6	2^{32}	10	2^{51}
3	2^{17}	7	2^{37}	11	2^{55}
4	2^{22}	8	2^{40}	12	2^{63}

Table 4: Estimated number of chosen plaintext pairs for the differential attack described in §8 on RC5 with 64-bit block size.

We implemented the attack for $w = 32, r \leq 6$ on a Sun4 workstation. The actual number of plaintexts used matched the theoretical calculation, and the success rate was very high. Note that for each S_i, only 64 plaintext/ciphertext pairs were actually used for computing the key, and all other pairs were discarded immediately after they were generated. In addition, no exhaustive search is needed in the attack. Therefore, in the implementation, the time used for computing the S table was negligible (less than a second on the Sun4) after sufficient good pairs were generated.

8.2 Improved differential attacks on RC5

In the preceding section, we described the details of the first differential attack on RC5 by Kaliski and Yin [7]. In this section, we will summarize the main ideas in the two subsequent improved differential attacks on RC5 by Knuden and Meier [9] and by Biryukov and Kushilevitz [3].

Knudsen and Meier's attack

In Knudsen and Meier's attack, the characteristics used for the "inner" rounds of RC5 are the same as those in Kaliski and Yin's attack. For

the rounds at the beginning and at the end of the cipher, however, more complicated characteristics are derived by analyzing the relations among input, output, and the subkeys. More specifically, they make the following two insightful observations.

First, if the least significant $\lg w$ bits of both halves of the plaintext are chosen to have appropriate values (which are dependent on the subkeys), then the two rotation amounts in the first full round of RC5 will be zero. In other words, by imposing additional constraints on a pair of plaintexts, the difference can propagate through the first full round with much higher probability compared with the corresponding characteristic in the early attack. It is also showed that detecting such appropriate constraints can be done fairly efficiently.

Second, the last-round characteristic (Ω^4, Ω^5) used in Kaliski and Yin's attack (see §8.1.2) is just one possible characteristic for detecting a good pair, and it is one with small Hamming weights. In general, the Hamming weights of the differences in the last few rounds may follow a pattern similar to a Fibonacci sequence. And such a relaxation for the constraints on characteristics in the last few rounds also yield characteristics with higher probabilities.

By combining these two observations, a factor of up to 2^9 reduction in the plaintext requirements can be obtained when compared with Kaliski and Yin's attack.

Knudsen and Meier also consider certain "differentially weak keys" of RC5 with respect to their attack. They showed that for a small portion of the keys ($2^{-5.37t}$, for $t \geq 1$), their attack can be further enhanced by a factor of approximately 2^{2t}.

Biryukov and Kushilevitz's attack

Biryukov and Kushilevitz consider more complex characteristics than those used in the previous attacks and define a more general notion of good pairs with respect to data-dependent rotations. In particular, all plaintext/ciphertext pairs that escape differences in rotation amounts can be used in their attack, not just pairs that follow specific patterns (e.g., see §8.1.2). It is not hard to see that such characteristics occur with much higher probability than the one-bit characteristics. They also generalize the above mentioned observations of Knudsen and Meier by introducing the concepts of "space oracles" and "corrected Fibonacci sequences."

Roughly speaking, a space oracle is a partition of the set of all possible plaintexts such that certain subsets of the partition have a much higher density of good pairs than other subsets. So a space oracle is a generalization

of the first observation made by Knudsen and Meier, and it allows good pairs to be found in fewer steps than by searching through the entire set of plaintexts. Biryukov and Kushilevitz derive efficient space oracles for which the differences in a pair of plaintexts can pass through two and a half rounds at the beginning of the cipher with very high probability.

Corrected Fibonacci sequences more accurately model how the Hamming weights of the differences propagates for a given good pair, since differences can sometimes be canceled (and hence Hamming weights can be reduced) due to the exclusive-or operation in the round function of RC5. Biryukov and Kushilevitz experimentally generated all possible Fibonacci sequences for all reasonable numbers of corrections up to 16 rounds, and the result gives a good theoretic estimate for the probability of a good pair. Such a model also provides a good method for finding good pairs by filtering the output difference.

The use of the above more sophisticated techniques yields an additional factor of up to 2^{10} reduction in the plaintext requirements over the improvements obtained in Knudsen and Meier's attack. Biryukov and Kushilevitz estimate that RC5 with 12 rounds and 64-bit block size can be attacked using about 2^{44} plaintexts.

8.3 The limitations of differential cryptanalysis on RC5

Recall that in the differential cryptanalysis of RC5, we use only half-round characteristics for which the pair of inputs have the same rotation amounts (i.e., $R'_{i-1} \bmod w = 0$). Such a choice for characteristics is based on the following intuition: If the pair of inputs have different rotation amounts in a characteristic, then the pair of outputs can be expected to differ in many possible ways, and so the characteristic will not be useful in a differential attack.

To give an analytical justification of the above intuition, we will take a closer look at the data-dependent rotations. First, for a pair of inputs (X, R) and (X^*, R^*), we define

$$\begin{aligned} Y &= X \lll R, \\ Y^* &= X^* \lll R^*, \\ X' &= X \oplus X^*, \\ Y' &= Y \oplus Y^*. \end{aligned}$$

For a give input difference X' and two rotation amounts R and R^*, we will analyze the distribution of the output difference Y' when X and X^*

range over all possible values. Let

$$\begin{aligned} D(X', R, R^*) &= \text{set of all possible values for } Y', \text{ and} \\ N(X', R, R^*) &= \text{number of distinct vectors in } D(X', R, R^*). \end{aligned}$$

Lemma 8.1 *Let* $r' = (R - R^*) \bmod w$ *and* $k = \frac{w}{gcd(w, r')}$. *Then* $N(X', R, R^*) = 2^{\frac{k-1}{k}w}$ *and each of the* $N(X', R, R^*)$ *distinct binary vectors occurs exactly* $\frac{2^w}{N(X', R, R^*)}$ *times in the set* $D(X', R, R^*)$.

Proof. We prove the lemma by analyzing the constraints imposed on a vector $y \in D(X', R, R^*)$. We first rewrite y as follows:

$$\begin{aligned} y &= (X \lll R) \oplus (X^* \lll R^*) \\ &= (X \lll R) \oplus (X \lll R^*) \oplus (X' \lll R^*) \end{aligned}$$

Therefore, for $0 \le i \le w - 1$,

$$y[i] = X[(i - R) \bmod w] \oplus X[(i - R^*) \bmod w] \oplus X'[(i - R^*) \bmod w].$$

Consider the special case where r' is odd. The only constraint imposed on y is

$$parity(y) = parity(X').$$

Hence, the number of different y's is 2^{w-1} and each one occurs exactly twice. The general case can be analyzed similarly. □

In what follows, we consider some implications of Lemma 8.1 by contrasting the case that $r' = (R - R^*) \bmod w = 0$ with the case $r' \neq 0$:

1. $r' = 0$. The input difference does not affect the rotation amount.

   In this case, we have $k = 1$ and $N(X', R, R^*) = 1$. In other words, there is only *one* possible output difference Y'. All the half-round characteristics used in the differential attack (see §8.1.1) belong to this case.

2. $r' \neq 0$. The input difference affects the rotation amount.

   In this case, k is a power of 2 and ranges between 2 (when $r' = w/2$) and w (when r' is odd). Hence, $N(X', R, R^*)$ ranges between $2^{\frac{w}{2}}$ and 2^{w-1}, and each of the different binary vectors occurs the same number of times. In other words, the output difference Y' is uniformly distributed in a set of at least $2^{\frac{w}{2}}$ possible values when the pair of inputs with a fixed difference ranges over all possible values.

From the above discussions, we can see that the difference in the input are spread out in a drastic way once the difference in a half-round affects the rotation amount. Clearly, the larger the Hamming weight in the difference, the higher chance that the difference will affect the rotation amounts. So a good characteristic for RC5 should keep the Hamming weights for the intermediate differences as small as possible.

8.4 Markov properties of RC5

Here we show that RC5 is not a *Markov cipher* with respect to either the exclusive-or "$\oplus$" difference the subtraction "$-$" difference. Then we argue that even though RC5 is not a Markov cipher, it has an important property of a Markov cipher which is useful for a cipher to be secure against differential cryptanalysis.

The notion of a Markov cipher was introduced by Lai, Massey, and Murphy [11], and it is a useful tool in analyzing the resistance of an iterative cipher to differential cryptanalysis. Loosely speaking, an iterative cipher is *Markov* if there is a way of defining differences such that the probability of an output difference of the round function depends only on the input difference and is independent of the values of inputs. It has been proved that both DES and IDEA are Markov ciphers[11].

If an iterative cipher is Markov and its round subkeys are independent, then the sequence of differences at each round output forms a Markov chain. Under certain assumptions, every output difference will be roughly equally likely after sufficiently many rounds. Hence, the cipher will be secure against a differential attack when the number of rounds is sufficiently large.

Lemma 8.2 *RC5 is not a Markov cipher with respect to exclusive-or.*

Proof. Let (L_{i-1}, R_{i-1}) and (L^*_{i-1}, R^*_{i-1}) be a pair of inputs to a half-round of RC5. If $R_{i-1} = R^*_{i-1} = 0$, then we have

$$\begin{aligned} R_i &= L_{i-1} + S_i, \\ R^*_i &= L^*_{i-1} + S_i. \end{aligned}$$

Let e_s denote the w-bit binary vector which is 1 in bit s and 0 everywhere else. If we set $L'_{i-1} = e_s$ for some $s < w - 1$, then $R'_i = e_s$ with probability $1/2$ for random key S_i. On the other hand, if $R_{i-1} = R^*_{i-1} = 1$, then we have

$$\begin{aligned} R_i &= (L_{i-1} \oplus 1) \lll 1 + S_i, \\ R^*_i &= (L^*_{i-1} \oplus 1) \lll 1 + S_i. \end{aligned}$$

When $L'_{i-1} = e_s$, the probability that $R'_i = e_s$ is zero since R_i and R^*_i will differ in bit position t with $t \geq s + 1$ because of the rotation. Thus, the probability of an output difference depends on the values of inputs, so RC5 is not Markov with respect to exclusive-or. □

Lemma 8.3 *RC5 is not a Markov cipher with respect to subtraction.*

Proof. Similar to the proof of Lemma 8.2. □

The main reason that RC5 is not a Markov cipher with respect to exclusive-or and subtraction is the data-dependent rotation. Furthermore, it is very unlikely that RC5 is a Markov cipher with respect to some complicated difference measure. (For most block ciphers, the difference measure is quite obvious. There is one exception though since IDEA is a Markov cipher with respect to an unusual difference measure.)

A Markov cipher has many properties, but the property that is important for the cipher to be secure against differential attack is that every output difference will be roughly equally likely after sufficiently many rounds. We have seen that RC5 is not a Markov cipher due to the use of data-dependent rotations. However, as we have previously discussed in §8.3, the output difference of Equation $Y = X \lll R$ is uniformly distributed over a large set of possible values if the input difference affects the rotation amounts. As the number of rounds increase, the probability that the input difference to a half-round will affect the rotation amounts approaches one. Even though it may not be the case that *every* output difference will occur, the large number of possible output differences would make a differential attack impossible.

In sum, RC5 is not Markov when we consider each single half-round, but RC5 with a sufficiently large number of rounds possesses a Markov-like property that is important for preventing a differential attack.

9 RC5 and Linear Cryptanalysis

In this section, we will study the security of RC5 against linear cryptanalysis. We will focus our discussions on how to construct linear approximations for RC5 based on the results in [7]. We will also consider ways of using these linear approximations to mount linear attacks on RC5 and some hidden assumptions that affect the success rate of such attacks [21].

As we will see, it seems to be much harder to mount a linear attack against RC5 than a differential attack. So later in the section, we will analyze how the mixed use of rotations and additions in RC5 helps prevent linear cryptanalysis.

9.1 Linear approximations for a half-round of RC5

In this section, we consider linear approximations for a half-round of RC5. We will say that a linear approximation is *perfect* if it holds with bias 1/2 (probability 1 or 0). (Note that this *perfection* is from the viewpoint of the attack!)

Recall that there are two equations in a half-round.

$$\begin{aligned} L_i &= R_{i-1}, \\ R_i &= ((L_{i-1} \oplus R_{i-1}) \lll R_{i-1}) + S_i. \end{aligned}$$

For the first equation, there are many trivial approximations which involve the same bits of L_i and R_{i-1} and hold with probability 1. For example,

$$L_i[0] = R_{i-1}[0].$$

Following notation that has been established in the literature [14], we will denote the above trivial approximation as `-`.

To find good linear approximations for the second equation, we decompose it into three equations, each of which involves only a single primitive operation, and we consider possible linear approximations for each of them.

$$\begin{aligned} X &= L_{i-1} \oplus R_{i-1}, \\ Y &= X \lll R_{i-1}, \\ R_i &= Y + S_i. \end{aligned}$$

The bias of an approximation for $R_i = Y + S_i$ is in general dependent on the subkey S_i. Consequently, the bias of an approximation for a half-round is also key-dependent. Throughout our discussions, we will use *average bias* over all possible subkeys as the measurement for the bias of an approximation of RC5. More precisely, given an approximation `A`, we define

$$\text{average bias of } \texttt{A} = (1/2^w) \times \sum_{S_i} (\text{bias of } \texttt{A} \text{ when subkey is } S_i).$$

Since the bias of `A` for subkey S_i is always non-negative, the average bias of `A` is also non-negative. The average bias appears to be a fairly easy to compute while useful measurement in the linear cryptanalysis of RC5 as well as other block ciphers. Similar to the average analysis for differential characteristics in §8.1.1, we assume that the subkeys will be essentially random and independent of one another given a good key expansion algorithm.

9.1.1 Analyzing individual operations

The exclusive-or operation

The equation $X = L_{i-1} \oplus R_{i-1}$ has numerous perfect linear approximations. In particular, all approximations involving the same bits of X, L_{i-1}, and R_{i-1} are perfect. All other approximations have zero bias.

The rotation operation

The linear approximations for the equation $Y = X \lll R_{i-1}$ can be divided into two types depending on whether bits of R_{i-1} are involved.

- No bits of R_{i-1} are involved.

  Any such approximation involving just one bit of X and Y holds with probability $1/2 + 1/2w$, since for one rotation amount, the bits are guaranteed to be equal and for the other $w - 1$ amounts, the bits will be equal with probability $1/2$ (assuming the inputs are random). In general, for $t = 0, \ldots, \lg(w)$, an approximation involving 2^t bits of X (spaced at $w/2^t$-bit intervals) and 2^t bits of Y (that is a rotation of X) holds with probability $1/2 + 2^t/2w$.

- Some bits of R_{i-1} are involved.

  Some of these approximations have a non-zero bias. For example,

  $$Y[0] = X[0] \oplus R_{i-1}[0] \tag{3}$$

  holds with probability $1/2 + 1/2w$, since when the rotation amount is zero, $R_{i-1}[0] = 0$ and $Y[0] = X[0]$. When the rotation amount is non-zero, the equation holds with probability $1/2$. We remark that an approximation will have zero bias if it involves any bits of $R_{i-1}[s]$ where $s \geq \lg(w)$.

The addition operation

The best linear approximation for the equation $R_i = Y + S_i$ is

$$R_i[0] = Y[0] + S_i[0], \tag{4}$$

which holds with probability 1 for any subkey S_i (so the average bias is $1/2$). All other approximations are not perfect. For example, the bias of the approximation $R_i[1] = Y[1]$ ranges from 0 to $1/2$ for different subkeys and is averaged at $1/4$. In general, the average bias gets smaller as more bits are involved in an approximation.

9.1.2 One-bit linear approximations

We can construct many possible linear approximations for a half-round of RC5 given the approximations for individual operations. To start with, we consider some one-bit linear approximations.

By joining $X[0] = L_{i-1}[0] \oplus R_{i-1}[0]$, Approximation (3), and Approximation (4), we obtain the following approximation for a half-round:

$$R_i[0] = L_{i-1}[0] \oplus S_i[0].$$

This approximation holds with probability $1/2 + 1/2w$ for any subkey S_i. We will denote it as `E`. Note that `E` has an average bias of $1/2w$ which is the *same* for any subkey S_i. (For simplicity, we will omit the word "average" when it is clear from the context.) A nice feature of `E` is that it can be alternated with the trivial approximation `-`.

For the first half-round which uses only the + operation, both approximations

$$L_1[0] = L_0[0] \oplus S_0[0] \quad \text{and} \quad R_1[0] = R_0[0] \oplus S_1[0]$$

hold with probability 1. We will denote them as `C` and `D`, respectively.

9.1.3 Multiple-bit linear approximations

Here we will consider some linear approximations for a half-round of RC5 that involve multiple bits. We will then compare the biases of these approximations with the biases of one-bit approximations.

For the $\lll$ operation, we consider approximations such that none of the bits of R_{i-1} is involved. Again, for $t = 0, ..., \log(w)$, an approximation involving $k = 2^t$ bits of X (with equal intervals) and 2^t bits of Y (that is a rotation of X) holds with bias $2^t/2w$. For example, for $w = 16$ and $t = 2$, the approximation $X[0, 4, 8, 12] = Y[1, 5, 9, 13]$ holds with bias $1/8$.

For the + operation, we need to match with the approximations for $\lll$ in order to cancel Y. So we choose the approximation for + that involves the same bits of Y and R_i as in the approximation for $\lll$ (e.g., $Y[1, 5, 9, 13] = R_i[1, 5, 9, 13]$).

Once we fix the approximations for $\lll$ and +, we choose the approximation for $\oplus$ that matches with the approximation (e.g., $X[0, 4, 8, 12] = L_{i-1}[0, 4, 8, 12] \oplus R_{i-1}[0, 4, 8, 12]$).

As the number of involved bits k in the approximation increases, the bias for $\lll$ increases and the average bias for + decreases. At first glance, it is not clear for which value k the approximation gives the largest average

bias. We did some preliminary experiments, and the results for word sizes $w = 4, 8, 16$ are provided in the following three tables.

$w = 4$.

# bits involved (k)	bias for $\lll$	average bias for $+$	total average bias
1	$1/8$	$1/2$	$2/16$
2	$1/4$	$\leq 1/4$	$\leq 2/16$
4	$1/2$	$3/16$	$3/16$

$w = 8$.

# bits involved (k)	bias for $\lll$	average bias for $+$	total average bias
1	$1/16$	$1/2$	$32/2^9$
2	$1/8$	$\leq 1/4$	$\leq 32/2^9$
4	$1/4$	$\leq 27/2^8$	$\leq 27/2^9$
8	$1/2$	$18/2^8$	$36/2^9$

$w = 16$.

# bits involved (k)	bias for $\lll$	average bias for $+$	total average bias
1	$1/32$	$1/2$	$2048/2^{16}$
2	$1/16$	$\leq 1/4$	$\leq 2048/2^{16}$
4	$1/8$	$\leq 4368/2^{16}$	$\leq 1092/2^{16}$
8	$1/4$	$\leq 1074/2^{16}$	$\leq 537/2^{16}$
16	$1/2$	$608/2^{16}$	$608/2^{16}$

We notice that for word sizes $w = 4, 8$, the approximations that involve all w bits have the largest average bias. However, experiments also showed that for word sizes $w = 16, 32, 64$ (as proposed in [17]), the approximations that involve one bit have the largest average bias.

9.2 Linear approximations of RC5

Given the linear approximations for a half-round of RC5 in §9.1, it is quite easy to construct linear approximations for RC5 with any number of rounds.

We again start with one-bit linear approximations. It is easy to see that `D-E-E-...E-` is a linear approximation for i half-rounds if i is even, and `CE-E-...E-` is a linear approximation for i half-rounds if i is odd. For $n-1=2r$ half rounds, the approximation `D-E-E-...E-` may be written as

$$R_0[0] \oplus L_{n-1}[0] = T_n, \tag{5}$$

where

$$T_n = S_1[0] \oplus S_3[0] \oplus \cdots \oplus S_{n-2}[0]$$

is a fixed key bit for a given expanded key table S.

Since `E` appears exactly $(n-1-2)/2 = r-1$ times, by Matsui's "piling-up" lemma[4] [14], Approximation (5) holds with probability $1/2 + 1/2w^{r-1}$. As a consequence, the bit $R_0[0] \oplus L_{n-1}[0]$ is biased toward T_n.

In general, we can also construct linear approximations for RC5 that involve multiple bits, but as it is discussed in §9.1.3, such approximations would have smaller biases than one-bit approximations for the intended block sizes.

9.3 Implementing the linear attack

In this section, we discuss two approaches of using Approximation (5) to mount a linear attack on RC5 and some issues in an actual implementation.

A fairly straightforward approach would be to follow standard techniques in linear cryptanalysis. More specifically, the basic idea is to try each of the 2^{32} possible subkeys S_n, considering the one that yields the largest experimental bias for Approximation (5) to be the correct key. However, it is possible that many guesses for S_n may yield essentially the same bias where the wrong guesses can only be ruled out after unwrapping several rounds. Therefore, the work effort for this attack could be much more than 2^{32}, and experiments are needed to correctly estimate the actual work effort. Again following standard techniques, the plaintext requirement for this attack is approximately equal to the inverse square of the bias, that is, $4w^{2(r-1)}$.

A more sophisticated approach follows the general method on attacking RC5 outlined in §7. The basic idea is to first obtain the key bit T_n and

[4] It has been shown [5] that the "piling-up" lemma may not be applied to certain definitions of average bias. In our case, since the average bias of `E` is the same for all keys, the lemma can be applied.

then use Approximation (5) to approximate $L_{n-1}[0]$ for each given plaintext/ciphertext pair. As discussed in §7, the bit $L_{n-1}[0]$ will then allow one to compute the subkey S_n using Equation (2). The details of this attack were presented in [7], and it was estimated that the success rate of the attack is around 90% with $4w^{2(r-1)}$ plaintexts. However, Selcuk [21] later discovered that the actual success rate of the attack was only around 10-15% due to certain hidden assumptions.

In particular, since Equation (2) is derived for each fixed rotation amount $R_{n-1} \bmod w = s$, to use Approximation (5) together with Equation (2), the following assumption is needed:

- , *Assumption R*: For $s = 0, ..., w-1$, Approximation (5) holds with probability (approximately) $1/2 + 1/2w^{r-1}$ for randomly chosen plaintext/ciphertext pairs such that $R_{n-1} \bmod w = s$,

Preliminary experiments reported in [21] showed that the bias varied for each value of s. More analysis and experiments are still on going to fully determine the plaintext requirements in more sophisticated linear cryptanalytic attacks.

9.4 The limitations of linear cryptanalysis on RC5

It is interesting to consider the limitations of linear cryptanalysis on RC5 by analyzing how the mixed used of operations help prevent from constructing good linear approximations.

From the discussions in §9.1, we can see that the rotation and addition operations are incompatible when trying to find linear approximations for a half-round that have the largest average bias: the bias gets larger for $\lll$ if more bits are involved in an approximation, and the average bias gets smaller for $+$ if more bits are involved. Preliminary experiments give strong evidence that for $w = 16, 32, 64$, approximation `E` has the largest average bias among all approximations for a half-round (see Appendix) for more details).

We thus conjecture that for the word sizes $w = 16, 32, 64$ proposed in [17], linear approximation `DE-E-...` has the *largest average bias* among all approximations for RC5. If the conjecture holds, we would then be able to conclude that standard linear cryptanalysis is only effective for RC5 with a very small number of rounds.

In addition to the experimental evidence, we also have analytical evidence for the correctness of the conjecture. In particular, we show that

E is a best half-round approximation that can be alternated with a trivial approximation.

Lemma 9.1 *Let set M contain all half-round approximations in which neither bits of R_{i-1} nor bits of L_i are involved. Then* E *has the largest average bias among all approximations in M.*

Proof. Let F be an arbitrary approximation in M. Then F can be decomposed into three approximations, one for each operation. There may be many possible decompositions, and we consider the constraints on the three approximations for a given decomposition. The approximation for $Y = X \lll R_{i-1}$ cannot involve $R_{i-1}[s]$ with $s \geq \lg(w)$ since F has bias zero otherwise. Hence, the approximation for $X = L_{i-1} \oplus R_{i-1}$ cannot involve $X[s]$ with $s \geq \lg(w)$; otherwise, either F involves bits of R_{i-1} or it has bias zero. Any approximation for $Y = X \lll R_{i-1}$ involving only $X[s]$ with $s \leq \lg(w) - 1$ holds with bias at most $1/2w$ since there is only one rotation amount that can match the bit positions of X and Y. Therefore, F has bias at most $1/2w$. Since E holds with bias $1/2w$, it is a best approximation among all approximations in M. □

In sum, both experimental and analytical results show that the mixed use of rotation and addition operations provides good security for RC5 against linear cryptanalysis.

10 Further Considerations

10.1 Exhaustive search attack on RC5

We know that the security of a block cipher against exhaustive search is closely related to the key size used in the block cipher. The secret key used in RC5 has a variable length b with allowed values range from 0 to 255 bytes, and the expanded key table for RC5 with r rounds has $2^{(2r+2)w}$ bits for the $2w$-bit block size. So the effort for a brute-force attack on RC5-$w/r/b$ is $\min\{2^{8b}, 2^{(2r+2)w}\}$. Hence, if both the length of the secret key and the number of rounds are sufficiently large, RC5 is secure against exhaustive search.

Unlike DES, which has no parameterization and hence no flexibility in the security against exhaustive search, RC5 permits upgrades as necessary. For example, one can easily upgrade RC5 with 56-bit key to an 80-bit key. As technology improves, and as the true strength of RC5 algorithms becomes better understood through analysis, the most appropriate parameter values can be chosen.

r	$N_r(31)$	r	$N_r(31)$
1	74,464,461	5	99,998,944
2	96,489,501	6	99,999,953
3	99,709,954	7	99,999,996
4	99,981,305	8	100,000,000

Table 5: A statistical test for the rotation operation. In the table, $N_r(31)$ denotes the total number plaintexts in 100 million random plaintexts for which flipping bit 31 of the plaintext results in changes in some rotation amount within r rounds.

In January 1997, RSA Laboratories has launched the *RSA Data Security Secret-Key Challenge* [20] for both DES and RC5, in the hope that the resistance of ciphers to exhaustive key search attacks can be more accurately gauged in the future. For each contest, the unknown plaintext message is preceded by three known blocks of text that contain the 24-character phrase "The unknown message is: ". While the mystery text that follows will clearly be known to a few employees of RSA Data Security, the secret key itself used for the encryption was generated at random and never revealed to the challenge administrators. The goal of each contest is for participants to recover the secret randomly-generated key that was used in the encryption.

As of this writing, the challenges for RC5 with a 40-bit key, 48-bit key and 56-bit key have already been solved [20]. It took 3.5 hours for the 40-bit challenge, 313 hours for the 48-bit challenge, and 265 days for the 56-bit challenge, as was expected. It is anticipated, however, that some of the longer key lengths (80 bits or more) will remain an unsolved challenge for some considerable time to come.

10.2 Statistical analysis of RC5

Statistical analysis of RC5 for both the key expansion routine and the encryption routine has been one of the ongoing project. So far we have performed a series of standard statistical analysis including *the frequency test, the serial test, the poker test, the run test,* and *the auto-correlation test* for a selection of key sizes and number of rounds. Early results show that RC5 has good statistical characteristics.

Here we present the results of one special statistical test that examines how fast a difference in a pair of plaintexts will result a difference in the rotation amounts as the number of rounds increases. As we pointed out earlier, the heavy use of data-dependent rotations is one distinguished features of RC5, and hence it is important to know how this feature affects the cipher statistically.

More specifically, we performed the following test. In 100 million ($\approx 2^{23}$) trials with random plaintext and keys, we checked whether a pair of plaintexts differing in a single bit lead to some different intermediate rotation amounts. For RC5-32/r (64-bit block size, r rounds), let $N_r(s)$ denote the total number of such pairs in 100 million trials when bit s of the plaintext is flipped. Table 5 lists the value of $N_r(31)$ for increasing r.

For other values of s, $N_r(s)$ increases (as r increases) at a faster rate than $N_r(31)$. Overall, we found that with very high probability, flipping an input bit would affect some rotation amount for RC5-32 with eight rounds.

10.3 Modified versions of RC5

In the analysis of a cipher, it is often very instructive to consider the resistance of some cipher variant to cryptanalytic attacks. This often gives some insight to the security of the real cipher. So in this section, we consider some modified versions of RC5. We try to analyze the strength and weakness of each new version compared to RC5. We name the modified versions in a certain way just for ease of reference.

RC5XOR: $R_i = ((L_{i-1} \oplus R_{i-1}) \lll R_{i-1}) \oplus S_i$

RC5XOR is less secure than RC5 against both differential and linear cryptanalysis. In particular, the change of $+$ to $\oplus$ increases the probability of a half-round characteristic by a factor of about 2^t if the Hamming weight of the characteristic is t. Nevertheless, existing results [3] showed that RC5XOR serves as a good starting point for one to analyze RC5 since it preserves the basic structure of RC5 while only requiring a smaller number of plaintexts to mount the same attack.

RC5P: $R_i = ((L_{i-1} + R_{i-1}) \lll R_{i-1}) + S_i$

The change of $\oplus$ to $+$ reduces the probability of some half-round characteristics by a small factor if exclusive-or is used as measure of difference. However, since addition is used twice, one can simply choose integer subtraction as the measure of difference, and so the strength of RC5P is comparable to RC5XOR against differential attacks. RC5P and RC5 seem to have the same security against linear attacks.

RC5PFR: $R_i = ((L_{i-1} \oplus R_{i-1}) \lll r_i) + S_i$, where r_i is a fixed rotation amount. The value r_i might be made public as a parameter of the cipher.

Even though the existing differential or linear attacks do not work well on RC5PFR due to its fixed rotation amounts, RC5PFR does not appear to be a strong cipher. In particular, starting with a given input difference, the only uncertainty in the evolution of differences is the carry effect. Therefore, there exist characteristics that hold with fairly high probability.

RC5KFR: $R_i = ((L_{i-1} \oplus R_{i-1}) \lll r_i(K)) + S_i$, where $r_i(K)$ is a rotation amount derived from the secret key K. In other words, the rotation amounts are key dependent and fixed for a given key.

For RC5KFR, if the attacker can guess the correct rotation amounts in each round, then the cipher reduces to RC5PFR. This requires about 2^{10r} guesses, and hence it may not be feasible for large r. The existing differential or linear attacks do not seem to apply to RC5KFR. However, since the rotation amounts are fixed, there might be some shortcuts for attacking the variant that we are not aware of at this point.

RC5RA: $R_i = ((L_{i-1} \oplus R_{i-1}) \lll f(R_{i-1})) + S_i$, where $F(R_{i-1})$ depends on all bits of R_{i-1} (not just the least significant five bits).

Since all existing differential attacks on RC5 use characteristics for which the pair of inputs both have the same rotation amount, the same attacks on RC5 would become less ineffective on RC5RA. Potentially, RC5RA may be a very strong cipher in light of our discussions in §8.3 about how data-dependent rotations provide a systematic way of preventing differential cryptanalysis.

There are different ways of realizing RC5RA. One possibility would be to modular reduce R_i by some small carefully chosen odd number, and another possibility would be to multiply R_i by some carefully chosen odd w-bit word and use the high order bits as the rotation amount. Both approaches would slow down the round function of RC5. Nevertheless, the increase in strength in each round makes it possible to reduce the number of rounds so that the overall speed of the cipher will remain the same as or perhaps even faster than the original RC5.

The recently proposed block cipher called RC6 [18] has adopted the above mentioned idea of computing rotation amounts. In RC6, the rotation amounts are obtained by taking the top five bits of the quadratic function $f(x) = x(2x + 1) \bmod 2^{32}$. Early analysis [4] showed that the combination of multiplication with data-dependent rotation in RC6 is very effective in thwarting differential attacks.

Part III
Executive Summary

In this report, we have assessed the security of RC5 using standard techniques from differential and linear cryptanalysis. We have also summarized the known cryptanalytic results on RC5.

The results to date, building on one another to apply advanced forms of differential and linear attack, have been very encouraging. We observe that RC5 with 12 rounds and 64-bit block size give roughly the same security as DES against these attacks – 2^{44} chosen plaintext pairs for RC5 as opposed to 2^{43} known plaintexts for DES. The extra speed of RC5 allows one to use extra rounds, thereby providing an additional margin of safety. Based on the known results, we conclude that RC5 with 16 rounds and 64-bit block size can provide good security against existing analytical attacks.

With the cipher receiving considerable attention from cryptanalysts worldwide, a picture of the security offered by RC5 has been quick to develop. Acceptance of the cipher is growing, and RC5 has been discussed for inclusion in various standards efforts and has been published by the IETF in RFC2040 [1]. Three years on, it seems that the RC5 block cipher offers a computationally inexpensive way of providing secure encryption.

We emphasize again two distinguishing features of RC5. The first feature is the heavy use of data-dependent rotations. Our analysis shows that data-dependent rotations are helpful for preventing differential and linear cryptanalysis. The second feature is the exceptional simplicity of the cipher, with the objective of making analysis easier. As we have seen, most of the characteristics and linear approximations for RC5 were derived analytically without any experimental search.

As of this writing, a new block cipher called RC6 [18], which is closely built on RC5, has been submitted to NIST for consideration as a candidate for the Advanced Encryption Standard (AES). Like RC5, RC6 makes essential use of data-dependent rotations and maintains simplicity in its design. We hope that the simple design of RC5 will help fully determine its security – and the security of ciphers derived from it – in a rapid way.

Acknowledgments

We would like to thank Bob Baldwin, Scott Contini, Ron Rivest, Matt Robshaw, and Ali Selcuk for helpful discussions.

References

[1] R. Baldwin and R. Rivest. *RFC 2040: The RC5, RC5-CBC, RC5-CBC-Pad, and RC5-CTS Algorithms.* October 30, 1996. Available at `ftp://ds.internic.net/rfc/rfc2040.txt`.

[2] E. Biham and A. Shamir. *Differential Cryptanalysis of the Data Encryption Standard.* Springer-Verlag, 1993.

[3] A. Biryukov and E. Kushilevitz. *Improved Cryptanalysis of RC5.* In *Advances in Cryptology — Eurocrypt '98*, pages 85–99, Springer, 1998.

[4] S. Contini, R.L. Rivest, M.J.B. Robshaw and Y.L. Yin. The Security of the RC6 Block Cipher. v1.0, August 20, 1998. Available at `www.rsa.com/rsalabs/aes/`.

[5] C. Harpes, G.G. Kramer, and J.L. Massey. A generalization of linear cryptanalysis and the applicability of Matsui's piling-up lemma. In L.C. Guillou and J.-J. Quisquater, editors, *Advances in Cryptology — Eurocrypt '95*, pages 24–38, Springer, 1995.

[6] B.S. Kaliski Jr. and M.J.B. Robshaw. Linear cryptanalysis using multiple approximations. In Y.G. Desmedt, editor, *Advances in Cryptology — Crypto '94*, pages 26–39, Springer, 1994.

[7] B.S. Kaliski Jr. and Y.L. Yin. On differential and linear cryptanalysis of the RC5 encryption algorithm. In D. Coppersmith, editor, *Advances in Cryptology — Crypto '95*, pages 171–183, Springer, 1995.

[8] B.S. Kaliski Jr. and Y.L. Yin. Data-dependent rotations help prevent differential cryptanalysis. Technical note, RSA Laboratories, August 1996.

[9] L.R. Knudsen and W. Meier. Improved differential attacks on RC5. In N. Koblitz, editor, *Advances in Cryptology — Crypto '96*, pages 216–228, Springer, 1996.

[10] P.C. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In N. Koblitz, editor, *Advances in Cryptology — Crypto '96*, pages 104–113, Springer, 1996.

[11] X. Lai, J.L. Massey, and S. Murphy. Markov ciphers and differential cryptanalysis. In D.W. Davies, editor, *Advances in Cryptology — Eurocrypt '91*, pages 17–38, Springer-Verlag, 1991.

[12] S.K. Langford and M.E. Hellman. Differential-linear cryptanalysis. In Y.G. Desmedt, editor, *Advances in Cryptology — Crypto '94*, pages 17–25, Springer, 1994.

[13] M. Matsui. The first experimental cryptanalysis of the Data Encryption Standard. In Y.G. Desmedt, editor, *Advances in Cryptology — Crypto '94*, pages 1–11, Springer, 1994.

[14] M. Matsui. Linear cryptanalysis method for DES cipher. In T. Helleseth, editor, *Advances in Cryptology — Eurocrypt '93*, pages 386–397, Springer, 1994.

[15] S. Moriai, K. Aoki, and K. Ohta. Key-dependency of linear probability of RC5. March 1996. To appear in *IEICE Trans. Fundamentals.*

[16] National Institute of Standards and Technology (NIST). *FIPS Publication 46-2: Data Encryption Standard.* December 30, 1993.

[17] R.L. Rivest. The RC5 encryption algorithm. In *Proceedings of the 2nd Workshop on Fast Software Encryption,* pages 86–96, Springer, 1995.

[18] R.L. Rivest, M.J.B. Robshaw, R. Sidney, and Y.L. Yin. The RC6 Block Cipher. v1.1, August 20, 1998. Available at `http://www.rsa.com/rsalabs/aes/`.

[19] M.J.B. Robshaw. *Block Ciphers.* Technical Report TR-601, version 2.0, RSA Laboratories, July 1995.

[20] *The RSA Data Security Secret-Key Challenge.* `<http://www.rsa.com/rsalabs/challenge97>`.

[21] A. A. Selcuk. *New Results in Linear Cryptanalysis of RC5.* In *Proceedings of the 5th Workshop on Fast Software Encryption,* pages 1–16, Springer, 1998.