A Timing Attack on RC5

Helena Handschuh† and Howard M. Heys‡

†ENST, Computer Science Department
46, rue Barrault, F-75634 Paris Cedex 13
handschu@enst.fr
†GEMPLUS, Cryptography Department
34, rue Guynemer, F-92447 Issy-les-Moulineaux
handschuh@gemplus.com
‡Memorial University of Newfoundland, Faculty of Engineering
St. John's, NF, Canada A1B 3X5
howard@engr.mun.ca

Abstract. This paper describes a timing attack on the RC5 block encryption algorithm. The analysis is motivated by the possibility that some implementations of RC5 could result in the data-dependent rotations taking a time that is a function of the data. Assuming that encryption timing measurements can be made which enable the cryptanalyst to deduce the total amount of rotations carried out during an encryption, it is shown that, for the nominal version of RC5, only a few thousand ciphertexts are required to determine 5 bits of the last half-round subkey with high probability. Further, it is shown that it is practical to determine the whole secret key with about 2^{20} encryption timings with a time complexity that can be as low as 2^{28}.

Keywords. Cryptanalysis, Timing Attacks, Block Cipher.

1 Introduction

RC5 is an iterative secret-key block cipher invented by R. Rivest [1]. It has variable parameters such as the key size, the block size, and the number of rounds. A particular (parameterized) RC5 algorithm is designated as RC5-$w/r/b$ where w is the word size (one block is made of two words), r is the number of rounds, and b is the number of bytes for the secret key. Our attack works for every choice of these parameters. However, we will focus on the "nominal" choice for the algorithm, RC5-32/12/16, which has a 64-bit block size, 12 rounds, and a 128-bit key.

The security of RC5 relies on the heavy use of data-dependent rotations. The application of the two powerful attacks of differential and linear cryptanalysis to RC5 is considered by Kaliski and Yin [2], who show that the 12-round nominal cipher appears to be secure against both attacks. In [3], Knudsen and Meier extend the analysis of the differential attacks of RC5 and show that, by searching for appropriate plaintexts to use, the complexity of the attack can be reduced by a factor of up to 512 for a typical key of the nominal RC5. As well, it is shown that keys exist which make RC5 even weaker against differential cryptanalysis. Recently, in [4], new differential cryptanalysis results imply that 16 rounds are required for the cipher with $w = 32$ to be secure. The results on linear cryptanalysis are refined by Selcuk in [5] and, in [6], it is shown that a small fraction of keys results in significant susceptibility to linear cryptanalysis. Despite these results, from a practical perspective RC5 seems to be secure against both differential and linear cryptanalysis.

In [7], Kocher introduces the general notion of a timing attack. The attack attempts to reveal the key by making use of information on the time it takes to encrypt. The applicability of the attack on asymmetric systems is demonstrated by examining timing variations for modular exponentiation operations. As noted by Kocher, implementations of RC5 on processors which do not execute the rotation in constant time are at risk from timing attacks. We will show that for implementations where the rotations take a variable amount of time, linear in the number of left shifts, RC5 is vulnerable to timing attacks which recover the extended secret key table with only 2^{20} ciphertexts from the sole knowledge of the total amount of rotations carried out during encryption.

2 Description of Cipher

RC5 works as follows: the secret key is first extended into a table of $2r+2$ secret words S_i of w bits. We will assume that this key schedule algorithm is rather one-way and will therefore focus on recovering the extended secret key table and not the secret key itself. The description of the key schedule can be found in [1].

Let (L_0,R_0) denote the left and right halves of the plaintext. Then the encryption algorithm is given by:

$$\begin{aligned}
&L_1 = L_0 + S_0 \\
&R_1 = R_0 + S_1 \\
&\text{for } i = 1 \text{ to } 2r \text{ do} \\
&\quad L_{i+1} = R_i \\
&\quad R_{i+1} = ((L_i \oplus R_i) \leftarrow R_i) + S_{i+1}
\end{aligned} \tag{1}$$

where "+" represents addition modulo-2^w, "$\oplus$" represents bit-wise exclusive-or, and "$X \leftarrow Y$" is the rotation of X to the left by the $\log_2 w$ least significant bits of Y. For example, if $w = 32$, X is rotated to the left by the number of positions indicated by the value of $Y \bmod 32$.

The ciphertext is (L_{2r+1},R_{2r+1}). The transformation performed for a given i value is called a half-round: there are $2r$ half-rounds. Each half-round involves exactly one data-dependent rotation and one sub-key S_i.

To decrypt, the operations of the algorithm must be appropriately reversed to generate the data for each half-round by essentially going backwards through the encryption algorithm. For example, data is rotated right and the subkeys are applied by subtracting modulo-2^w from the data.

In section 3 hereafter we describe the foundations of the timing attack and give some preliminaries and in section 4 we describe our timing attack as it is used to obtain $\log_2 w$ bits of the last half-round subkey. In section 5, we discuss how to derive the remaining subkey bits and, in section 6, we present some experimental results on the likelihood of the success of the attack. Finally, in section 7, we discuss the complexity of the attack.

3 Preliminaries

In this section, we describe our assumptions for the timing attack and show how to correlate the total amount of rotations carried out during encryption with the value of the second last rotation.

3.1 Timing attacks

For a complete description of Kocher's timing attacks we refer to [7]. The main idea is to correlate the time measurements for processing different inputs with the secret exponent in RSA or Diffie-Hellman like systems. Since for every non-zero bit in the secret exponent the whole process performs an extra modular multiplication which depends on some computable intermediate value, correlations can be found between the variations of the time measurements on some sample space and the fact that an exponent bit is set or not. This way, as the distributions become more accurate, more and more bits can be derived and finally the whole secret key can be recovered.

In symmetric-key cryptosystems, things tend to get more complicated as usually only constant-time operations such as additions, exclusive-ors or table look-ups are performed. Nevertheless, under certain assumptions, given cryptosystems like RC5 can become vulnerable to timing attacks as well.

3.2 Hypothesis

Rivest notes that "on modern microprocessors, a variable rotation ... takes *constant time*" but there are certain types of processors which do not have this property. For instance for 8-bit microcontrollers on smart cards, or in other constrained environments, the rotation has to be performed step by step, one left or right shift at a time. It is not necessarily optimal to swap bytes or nibbles depending on the number of left shifts, as testing the rest modulo 16, 8 or 4 of this number may take more time than doing all the shifts no matter what. When machine cycles are measured during encryption, we can deduce from a certain amount of measurements how long the constant-time operations take, and how long the variable-time operations take: as these only concern rotations for RC5, we can deduce what the total amount of rotations is for every encryption. We believe a specific analysis can also endanger the algorithm if rotations are carried out in a non-linear but still variable amount of time. As well, it should also be noted, that a naive, straightforward hardware implementation could also result in rotation times that are a function of the cipher data and, hence, create susceptibility to the timing attack we describe in this paper.

In this paper, we focus on the case where we can assume that a left rotation by an amount y takes y times longer than a left rotation by an amount of 1. We show a ciphertext-only attack that recovers the extended secret key in a reasonable amount of time when the total number of rotation amounts is given. Kocher has already mentioned that "RC5 is at risk" and we show why. Note that the imperfectness and inherent randomness of timing measurements may cause the complexity to grow and the required number of ciphertexts to be much higher, but as the theoretical attack we present is of such low complexity, it should raise serious doubts about the strength of RC5, if implemented such that the rotation times are data dependent.

3.3 Foundation of the Attack

Let T_{2r} denote the total rotation amount for the encryption of a given plaintext. T_{2r} is given by:

$$T_{2r} = \sum_{i=1}^{2r} (R_i \bmod w) \tag{2}$$

We note that the amount of the last rotation is known because it is the value of the $\log_2 w$ least significant bits of L_{2r+1} which is the left half of the ciphertext. Therefore let us consider the total amount of rotations minus the last rotation. We denote this quantity by T_{2r-1} and

$$T_{2r-1} = \sum_{i=1}^{2r} (R_i \bmod w) - (L_{2r+1} \bmod w). \tag{3}$$

More generally, for a half-round k, we can define the intermediate amount of rotations so far and we have

$$T_k = \sum_{i=1}^{k} R_i \bmod w \tag{4}$$

and

$$T_{k-1} = \sum_{i=1}^{k} R_i \bmod w - (L_{k+1} \bmod w). \tag{5}$$

We also have the following:

$$T_{k-1} = T_{k-2} + R_{k-1} \bmod w \tag{6}$$

Now let us consider the way the total amounts of rotations are distributed over some large sample space. T_{k-2} can be represented by its mean value $(k-2) \times \frac{w-1}{2}$ added to some deviation. If this deviation (noise) is small, T_{k-1} and $R_{k-1} \bmod w$ are correlated at each half-round and the distribution of T_{k-1} gives us a good idea about what the distribution of $R_{k-1} \bmod w$ should look like.

This is the observation that leads to our timing attack. Knowledge of the second last rotation amount gives us some knowledge about the last subkey. We describe the attack in detail in section 4 hereafter.

4 Our Attack

In this section we describe the first part of our attack, which is how to derive the $\log_2 w$ least significant bits of the last subkey, S_{2r+1}. We shall describe two approaches to extracting the $\log_2 w$ least significant bits of the last subkey. We shall refer to these approaches as Method A and Method B. (Both methods were derived independently and are further elaborated in [8] for Method A and [9] for Method B.) The fundamental difference between the approaches is the indicator used as a metric to pick the most likely subkey.

4.1 Method A

In the last section, we showed how the total amount of rotations T_{k-1} and a given rotation $R_{k-1} \bmod w$ are correlated. Now we need an indicator to be able to qualify this correlation. A quite natural choice consists in using the following I correlation coefficient as an indicator:

$$I = E\{(T_{k-1} - \mu_{k-1})(R_{k-1} \bmod w - \frac{w-1}{2})\} \tag{7}$$

where μ_{k-1} is the mean of the distribution of the T_{k-1}s over some sample space.

In fact, it is even more convenient to use only the sign of $(R_{2k} \bmod w - \frac{w-1}{2})$ in order to partition the samples depending on the deviation from their mean value. The indicator we shall therefore be using is the following:

$$I = E\{(T_{k-1} - \mu_{k-1}) \times Sign(R_{k-1} \bmod w - \frac{w-1}{2})\} \tag{8}$$

In the case of two correlated distributions, this indicator is expected to have a higher absolute value than in the case of two independent distributions (when our guess about the second last rotation amount is wrong).

The first phase of the attack is a sample generation phase. We collect triplets corresponding to a plaintext encrypted under the unknown key and its ciphertext, and T_{2r-1}, the total amount of rotations minus the last one carried out during encryption. These samples are stored in a table and are ordered by the value of the $\log_2 w$ least significant bits of the left half of the ciphertext. Recall that this also is the value of the last rotation amount.

We denote by N the number of collected samples of total rotation amounts in each category. At each half-round we assume that the intermediate rotation amounts are uniformly distributed, independent random variables. For our analysis to work, we need the standard deviation of our sample space to be negligible over all half-rounds and all samples.

Let X_i be the rotation amount of the i-th sample at an intermediate half-round. Over all half-rounds and all samples, we have:

$$Var(2r \times X_i) = 2r \times Var(X_i) = 2r \times Var(X_0) \tag{9}$$

and

$$Var(2r \times \sum_{i=1}^{N} X_i) = 2r \times N \times Var(X_0) \approx 2r \times N \times \frac{w^2}{12} \tag{10}$$

The standard deviation σ is given as:

$$\sigma^2 = Var(\mu) = Var(\frac{2r \times \sum_{i=1}^{N} X_i}{N}) \approx \frac{2r}{N} \times \frac{w^2}{12} \tag{11}$$

An order of magnitude of the number N of samples needed is given by the condition:

$$\sigma \ll 1 \tag{12}$$

which gives us the following condition for N:

$$N \gg 2r \times \frac{w^2}{12} \tag{13}$$

Therefore in each category we need N to be much greater than 2^{11}. From a practical point of view, we implemented our attack with 2^{15} samples in each category ; there are w different categories, therefore the total number of samples required is $2^{15} \times w = 2^{20}$. We will keep this upper bound in our complexity evaluation in section 7 as it is convenient for practical implementations. (Actually, the practical attack requires quite more time and plaintexts than suggested by the theory, so this approximation fits much better to the experiments.)

Now we have:

$$R_{2r+1} = ((R_{2r-1} \oplus L_{2r+1}) \leftarrow L_{2r+1}) + S_{2r+1} \tag{14}$$

In this last equation, R_{2r+1} is the right half of the ciphertext and L_{2r+1} is the left half of the ciphertext. Therefore the right value of R_{2r-1} gives us the right value of S_{2r+1}.

We concentrate on the category of samples such that the $\log_2 w$ least significant bits of the left half of the ciphertext are equal to zero. In particular, this means that the last rotation amount is also equal to zero. Therefore we have the following equation:

$$(R_{2r+1} - S_{2r+1}) \bmod w = R_{2r-1} \bmod w \tag{15}$$

For each possible value of the $\log_2 w$ least significant bits of S_{2r+1}, we compute the supposed second last rotation amount $R_{2r-1} \bmod w$ for each sample in the category we concentrate on. This gives us a trial distribution over the 2^{15} samples.

Now divide the samples into two parts depending on the sign of the guessed rotations. Compute the correlation coefficients I^+ and I^- on each of the two parts. On the negative part, the correlation coefficient is supposed to be negative, and on the positive part, it is supposed to be positive. Finally compute the quantity $I^+ - I^-$. This indicator should have a higher value when the two distributions are correlated then when they are independent. The right value of the $\log_2 w$ least significant bits of the last subkey is suggested by the highest indicator.

4.2 Method B

In this section, we describe another approach to extracting the $\log_2 w$ least significant bits of the last subkey and, using a probabilistic model we get an estimate of the number of ciphertexts required to determine the bits with high probability. For convenience, we shall strictly focus our attention on the cipher with $w = 32$ and $r = 12$. Similar to the previous section, the model assumes that the rotations in each half-round are independent random variables that are uniformly distributed over $\{0, 1, 2, \ldots, 31\}$ with a mean of 15.5 and a variance of 85.25. Under these assumptions, the sum of the number of rotations for the first 22 half-rounds of the cipher, T_{22}, is a random variable with a mean of $\mu_{22} = 22 \cdot 15.5 = 341$ and variance of $\sigma_{22}^2 = 22 \cdot 85.25^2 = 1875.5$. As well, based on the central limit theorem, T_{22} is approximately Gaussian distributed.

To determine the correct value of the partial subkey $S_{25} \bmod 32$, a number of ciphertexts is used to test each possible value for the 5 bits and to determine which value is most consistent with the expected statistical distribution of the timing information. In particular, ciphertexts for which $L_{25} \bmod 32 = 0$ are used to compute an estimate of the variance of T_{22} based on the value of each candidate partial subkey: it is expected that the variance estimate when the correct value for the partial subkey is selected will be smaller than the estimate when an incorrect partial subkey is assumed.

Let K represent the actual key bits $S_{25} \bmod 32$ and let $\tilde{K}$ represent the guess of the partial subkey K. The candidate key $\tilde{K}$ can be represented by

$$\tilde{K} = (K + \tau) \bmod 32 \tag{16}$$

where $-15 \le \tau \le 16$. The estimate of the variance of T_{22} for a particular candidate key $\tilde{K}$ is given by

$$\phi_\tau = E\left\{e_{\tau,x}^2\right\} \tag{17}$$

where $e_{\tau,x}$ represents the difference between the measured number of rotations for the entire 24 half-rounds and the expected number of rotations given the assumed candidate key for a ciphertext with $R_{25} \bmod 32 = x$. The difference $e_{\tau,x}$ is composed as follows:

$$e_{\tau,x} = (T_{22} + R) - (\mu_{22} + \tilde{R}_{\tau,x}) \tag{18}$$

where R represents the actual value of the rotation in the 23rd half-round (i.e., $R = R_{23} \bmod 32$) and $\tilde{R}_{\tau,x}$ represents the guess of the rotation in the 23rd half-round (corresponding to the candidate key $\tilde{K}$) given $R_{25} \bmod 32 = x$. Using ciphertexts for which $L_{25} \bmod 32 = 0$, the size of the rotation in the 24th half-round is 0. Therefore, $T_{22} + R$ equals the total number of rotations, which, of course, can be derived from the timing information. The value of $\tilde{R}_{\tau,x}$ is determined from $\tilde{R}_{\tau,x} = (x - \tilde{K}) \bmod 32$. Hence, for a given ciphertext and candidate key, the value of $e_{\tau,x}$ can be calculated. We can also view equation (18) by letting $\Delta T = T_{22} - \mu_{22}$ and $\Delta R_{\tau,x} = R - \tilde{R}_{\tau,x}$ and we get

$$e_{\tau,x} = \Delta T + \Delta R_{\tau,x}. \tag{19}$$

Now assume that the cryptanalyst has available N ciphertexts for which $L_{25} \bmod 32 = 0$ and, hence, for large N, the number of ciphertexts corresponding to a particular value x for $R_{25} \bmod 32$ is given by $N_x \approx N/32$. For each ciphertext and candidate key $\tilde{K}$, $e_{\tau,x}$ is computed and the mean of the square of $e_{\tau,x}$ is calculated. The result is equivalent to

$$\phi_\tau = \frac{1}{N} \sum_{x=0}^{31} \sum_{i=1}^{N_x} [\Delta T_{x,i} + \Delta R_{\tau,x}]^2 \tag{20}$$

where $\Delta T_{x,i}$ represents the i-th value of ΔT for $R_{25} \bmod 32 = x$. For the correct guess of the key (i.e., $\tau = 0$), $\Delta R_{\tau,x} = 0$ since $\tilde{R}_{\tau,x} = R$ and, hence, $\phi_0 = (1/N) \sum_{x=0}^{31} \sum_{i=1}^{N_x} (\Delta T_{x,i})^2$. For incorrect candidate keys, for which $|\tau| \geq 1$, $\Delta R_{\tau,x} \neq 0$, and it can be shown that $E\{\phi_\tau\} > E\{\phi_0\}$. The cryptanalyst can therefore collect ciphertexts and timing information (implying rotation information) and determine the key K by picking the candidate key $\tilde{K}$ which minimizes ϕ_τ.

We now determine the probability that an incorrect key is selected over the correct key. For this to happen, we must have $\phi_\tau < \phi_0$ or, alternatively, $\phi_\tau - \phi_0 < 0$. Hence, the cryptanalyst must acquire enough ciphertext timings to ensure that $\phi_\tau - \phi_0 > 0$ for all $\tau \neq 0$. From (20), it can be seen that

$$\phi_\tau - \phi_0 = \frac{1}{N} \sum_{x=0}^{31} \sum_{i=1}^{N_x} [2 \Delta R_{\tau,x} \Delta T_{x,i} + (\Delta R_{\tau,x})^2]. \tag{21}$$

We can consider $\phi_\tau - \phi_0$ to be a Gaussian distributed random variable with an expected value given by

$$E\{\phi_\tau - \phi_0\} \approx \frac{1}{32} \sum_{x=0}^{31} (\Delta R_{\tau,x})^2 \tag{22}$$

where we have used $N_x \approx N/32$ and $E\{\Delta T\} = 0$. As well, $\phi_\tau - \phi_0$ has a variance given by

$$Var\{\phi_\tau - \phi_0\} = \frac{1}{N^2} \sum_{x=0}^{31} \sum_{i=1}^{N_x} [(2 \Delta R_{\tau,x})^2 \sigma_{22}^2] \approx \frac{\sigma_{22}^2}{8N} \sum_{x=0}^{31} (\Delta R_{\tau,x})^2. \tag{23}$$

It can be shown that

$$\sum_{x=0}^{31} (\Delta R_{\tau,x})^2 = |\tau|(32 - |\tau|)^2 + (32 - |\tau|)|\tau|^2 = 32|\tau|(32 - |\tau|) \tag{24}$$

and, consequently, it can be easily verified that

$$\max_\tau P(\phi_\tau - \phi_0 < 0) = P(\phi_\omega - \phi_0 < 0) \tag{25}$$

where $\omega = -1$ or $+1$. For $N = 2000$ ciphertexts for which $L_{25} \bmod 32 = 0$, based on the Gaussian distribution, we get $P(\phi_1 - \phi_0 < 0) = 0.0021$ and the probability of being able to determine the correct 5 bits of subkey, $S_{25} \bmod 32$, is given by

$$P(S_{25}[4 \ldots 0] \text{ correct}) = 1 - P(\exists \tilde{K} | \tau \neq 0, \phi_\tau < \phi_0) \tag{26}$$

where

$$P(\exists \tilde{K} | \tau \neq 0, \phi_\tau < \phi_0) < 31 \cdot P(\phi_1 - \phi_0 < 0) = 0.0651. \tag{27}$$

Therefore, the probability of picking the correct 5 bits of subkey is greater than 93.5% with 2000 ciphertexts under the assumption that the rotations in all half-rounds are independent. Note that the ciphertexts must be chosen such that $L_{25} \bmod 32 = 0$, which is true on average for 1 in 32 random ciphertexts. Hence, the correct 5 bits of subkey can be derived with high probability using about 64000 random ciphertexts and their timings.

As we shall see in section 6, in fact, there are some dependencies in the rotations of different half-rounds which result in the probabilities of successfully deriving the key in practice being somewhat lower than expected from the model. Nevertheless, experimental evidence confirms that the approach works well when applied to the actual cipher.

5 Deriving the Remaining Subkey Bits

In the previous section, we illustrated how it is possible to determine $\log_2 w$ bits of the last half-round subkey S_{2r+1} with high probability using a set of random ciphertexts and their timing information. Fortunately, it is straightforward to apply the techniques on the same ciphertexts to determine the remaining bits of subkey S_{2r+1} and, with enough ciphertexts, it is possible to derive all the bits of the subkeys $S_i, 3 \leq i \leq 2r$, as well.

First, now that we have found the $\log_2 w$ least significant bits of S_{2r+1}, we have to derive the $w - \log_2 w$ other bits of the last subkey. We proceed the following way:

Based on the categories described in section 4.1, concentrate on one category of samples at a time, in increasing order. Depending on the value of the $\log_2 w$ least significant bits of the left ciphertext, the right half was rotated by some amount to the left. Therefore the right value of $R_{2r-1} \bmod w$ gives us the right value of the $\log_2 w$ bits of the last subkey which correspond to the $\log_2 w$ bit positions $i = [L_{2r+1} \bmod w]$ through $i = [L_{2r+1} \bmod w + \log_2 w - 1]$.

However, we proceed bit by bit in order to take advantage of the knowledge we already have on the least significant bits of the subkey. For each new rotation amount from 1 to $w - \log_2 w$, try each of the two possible values of the $\log_2 w$ concerned bits of the subkey: take the $\log_2 w - 1$ bits you already know and guess 0 or 1 for the next bit. This gives you only two possible distributions for $R_{2r-1} \bmod w$ in each category of samples. Using the indicator of either Method A or Method B determine the targetted key bit. The right value of the $\log_2 w$ concerned bits of the last subkey is suggested by the corresponding best indicator.

Once S_{2r+1} is derived, the remaining subkeys associated with each half-round $i, 2 \leq i \leq 2r - 1$, may be determined using the same set of ciphertexts. Once the subkey for a half-round i is determined, the ciphertext may be partially decrypted for one half-round so that the output of half-round $i-1$ is known. Correspondingly, the timing of the partial encryption of the first $i-1$ half-rounds may be determined by subtracting the time to execute the i-th half-round from the time to encrypt the first i half-rounds of the cipher. The new ciphertext and timing information may then be used to extract the subkey for half-round $i-1$ in exactly the same manner as for the subkey of half-round i.

The remaining subkeys, S_0, S_1, and S_2, are applied to the cipher by addition to the plaintext left half, plaintext right half, and the output of the rotation operation in the first half-round. All three of these subkeys cannot be determined using timing information but are trivially determined using only a modest number of known plaintexts and ciphertexts: S_1 is simply determined using one known plaintext and using the relationship $S_1 = L_2 - R_0$, S_2 can be determined with a modest number of known plaintexts using, for example, linear cryptanalysis [2], and S_0 can be easily derived once S_1 and S_2 are determined using $S_0 = [((R_2 - S_2) \rightarrow L_2) \oplus L_2] - L_0$.

6 Experimental Results

In this section we present the experimental results which validate the effectiveness of the attack. Both Methods A and B assume that the values of the rotations in different half-rounds are uniformly distributed, independent random variables. This assumption, however, is not strictly correct. Consider, for example, the following scenario for $w = 32$ and $r = 12$: $S_{24} = 0$ and $S_{25} \bmod 32 = 0$. Suppose the cryptanalyst is attempting to determine $S_{25} \bmod 32$ and is therefore considering ciphertexts for which $L_{25} \bmod 32 = 0$. If $R_{25} \bmod 32 = x = 16$, then $R_{22} \bmod 32 = (L_{25} \rightarrow 16) \oplus 16$ and, since $L_{25} \rightarrow 16$ is a uniformly distributed random variable, $R_{22} \bmod 32$ behaves as anticipated by the model. However, if $R_{25} \bmod 32 = x = 0$, then $R_{22} \bmod 32 = 0$ always and $R_{22} \bmod 32$ is not a uniformly distributed random variable as suggested by the model.

These discrepancies from the model add inaccuracies to the process of statistically deriving the subkeys for both Method A and Method B. However, experimental results for Method B demonstrate that the cryptanalytic technique is still very effective and the statistical model of section 4.2 provides a rough approximation of the effectiveness of the attack to determine 5 key bits of the last half-round subkey. Using Method B, for 2000 ciphertexts with $L_{25} \bmod 32 = 0$ (equivalent to about 64000 random ciphertexts), experiments on the nominal RC5 for 1000 random keys resulted in an 86.2% chance of the partial subkey $S_{25} \bmod 32$ being correctly determined. Using 64000 random ciphertexts, the complete subkey S_{25} was correctly determined for 69.7% of the keys.

The effectiveness of the timing attack as determined in experiments using Method B is further illustrated in Table 1. It is clear from the table that few random ciphertexts are required to determine the bits of the last half-round subkey with a high probability. The correct derivation of all subkeys $S_3 \ldots S_{25}$ does not occur with nearly as high a probability: even modest deviations in prob-

Number of	*Probability of Success*		
Random Ciphertexts	S_{25} mod 32	S_{25}	$S_3 \ldots S_{25}$
10^4	0.611	0.083	0.000
10^5	0.893	0.794	0.009
10^6	0.901	0.827	0.024

Table 1. Experimental Results for 1000 Keys on RC5-32/12/16 (Method B)

ability from 1 when determining subkeys significantly reduces the probability that all 23 subkeys will be successfully determined. However, it is apparent that the attack can be very effective in determining subkeys for a large fraction of keys and should be seriously considered to ensure that an implementation of the cipher is not vulnerable. Note that the probability of the complete success for the attack can be improved upon using a key path search approach described in the following section.

7 Complexity of the attack

In this section, we discuss the complexity of the attack in the context of Method A (although much of the discussion applies equally well to Method B). The first part of the attack is the sample generation phase. We need around 2^{15} ciphertexts in each category. There are w categories. Therefore the complexity of this phase is $2^{15} \times w$ encryptions. The ordering has no extra cost.

The second part of the attack is divided into four steps as mentioned in section 4. From a complexity point of view, there are $2r - 2$ half-rounds to be considered. For each half-round:

- start by computing the mean of the total rotation amounts in each category and by subtracting this mean to each total timing. This step takes $2^{15} \times w$ operations.
- then, computing the $\log_2 w$ least significant bits of a subkey takes $2^{15} \times w$ complexity.
- each further bit takes $2^{15} \times 2$ operations, and there are $w - \log_2 w$ such bits.
- finally, before entering the next half-round, decrypt one half of the ciphertext with the subkey just found, and reorder all the categories by the $\log_2 w$ least significant bits of this new half of the ciphertext. At the same time, subtract the value of the current last rotation from the total rotation amount for each ciphertext. This whole step takes $4 \times 2^{15} \times w$ operations.

Thus the total complexity for the last $2r - 2$ half-rounds is:

$$C = (2r - 2)[8 \times w \times 2^{15}] \tag{28}$$

As an example, for RC5-32/12/16 this complexity equals $C = 22 \times 2^{23}$. As for the last four subkeys, the cost is almost negligible.

In conclusion, the attack can be carried out with the encryption of only $2^{15} \times w$ plaintexts and the time complexity of the analysis phase is roughly equivalent to searching the last $2r - 2$ subkeys. The total complexity is:

$$C = (2r-2)[8 \times w \times 2^{15}] = (2r-2) \times 2^{18+\log_2 w} \tag{29}$$

These results were all checked by computer experiments based on Method A. In fact, the complexity is slightly higher because the right value of the $\log_2 w$ least significant subkey bits is not always suggested by the best indicators. Sometimes the second or third-best indicator corresponds to the right subkey. Thus, when implementing the attack, we had to make some complexity tradeoffs. However, when the top indicators are quite close, trying the 8 most likely paths is still possible. This only applies while searching the $\log_2 w$ least significant subkey bits; all other bits are always correctly guessed.

Experiments show that, using Method A, when the guess of the $\log_2 w$ least significant bits is wrong, along the next 4 half-rounds the indicators tend towards a characteristical pattern. Therefore the right key path can still be made out with little extra effort.

Considering RC5-32/12/16, for example, if there is no "wrong" indicator, the best complexity is around 2^{28} for one key search. On average, no more than 8 subkeys are expected to lead to a partial exhaustive path search. For every such key, searching through the 8 keys associated to the 8 best indicators over two half-rounds leads to the right result. Nevertheless, for the subkeys S_7 through S_4, this path search cannot apply anymore. Therefore we exhaustively search through the 8 best subkeys for these four half-rounds. The overall extra work factor should not exceed $8 \times 8^2 + 8^4 = 2^9 + 2^{12}$. However, this leads to an upper bound on the complexity which is far from being optimal.

In summary, the overall complexity of our attack does not exceed 2^{40} operations for RC5-32/12/16. On average, the complexity is quite lower though. It is more realistic to consider the best case where the analysis takes only 2^{28} operations.

In the general case, the complexity of the attack does not exceed:

$$C_{max} = (2r-2) \times 2^{30+\log_2 w} \tag{30}$$

On average it is closer to the theoretic complexity

$$C = (2r-2) \times 2^{18+\log_2 w} \tag{31}$$

and does not require more than $N_{max} = 2^{15+\log_2 w}$ ciphertexts and their associated timings, as well as a few known plaintexts to derive the last subkeys.

8 Conclusion

We have shown in some detail how to derive the extended secret key table of RC5-32/12/16 by a timing attack using only about 2^{20} ciphertext timings and in time complexity 2^{28} in the best case, and 2^{40} in the worst case. This confirms Kocher's statement that RC5 is at some risk on platforms where rotations take a variable amount of time, and suggests to be very careful when implementing RC5 on such platforms. Adding a random time to each encryption will not help as it will have little influence on the variance computations. Therefore we suggest to add the right amount of "dummy "rotations which will achieve a constant time for every encryption whatever the initial total amount of rotations.

9 Acknowledgements

We are very grateful to Henri Gilbert for the ideas and the help he gave us on this attack. We also would like to thank David Naccache for motivating this investigation, and Jacques Stern and Gérard Cohen for helpful comments.

References

1. R. L. Rivest. The RC5 Encryption Algorithm. In *Fast Software Encryption - Second International Workshop, Leuven, Belgium, LNCS 1008*, pages 86-96, Springer-Verlag, 1995.
2. B. S. Kaliski and Y. L. Yin. On Differential and Linear Cryptanalysis of the RC5 Encryption Algorithm. In *Advances in Cryptology - Crypto'95, LNCS 963*, pages 171-184. Springer-Verlag, 1995.
3. L. R. Knudsen and W. Meier. Improved Differential Attacks on RC5. In *Advances in Cryptology - Crypto'96, LNCS 1109*, pages 216-228, Springer-Verlag, 1996.
4. Biryukov and Kushilevitz. Improved Cryptanalysis of RC5. In *Advances in Cryptology - Eurocrypt'98, LNCS*, pages 85-99, Springer-Verlag, 1998.
5. A. A. Selcuk. New results in linear cryptanalysis of RC5. In *Fast Software Encryption - Fifth International Workshop, Paris, France, LNCS*, pages 1-16, Springer-Verlag, 1998.
6. H.M. Heys, Linearly Weak Keys of RC5. *IEE Electronics Letters*, vol. 33, no. 10, pp. 836-837, 1997.
7. Paul C. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In *Advances in Cryptology - Crypto'96, LNCS 1109*, pages 104-113, Springer-Verlag, 1996.
8. H. Handschuh, A Timing Attack on RC5. In *Workshop Record of SAC '98*, Queen's University, Kingston, Canada, pages 318-329, 1998.
9. H.M. Heys, A Timing Attack on RC5. In *Workshop Record of SAC '98*, Queen's University, Kingston, Canada, pages 330-343, 1998.