Linear Cryptanalysis of RC5 and RC6

Johan Borst, Bart Preneel, Joos Vandewalle

K.U. Leuven, Dept. Elektrotechniek-ESAT/COSIC
Kardinaal Mercierlaan 94, B-3001 Heverlee Belgium
Johan.Borst@esat.kuleuven.ac.be

Abstract. In this paper we evaluate the resistance of the block cipher RC5 against linear cryptanalysis. We describe a known plaintext attack that can break RC5-32 (blocksize 64) with 10 rounds and RC5-64 (blocksize 128) with 15 rounds. In order to do this we use techniques related to the use of multiple linear approximations. Furthermore the success of the attack is largely based on the linear hull-effect. To our knowledge, at this moment these are the best known plaintext attacks on RC5, which have negligible storage requirements and do not make any assumption on the plaintext distribution. Furthermore we discuss the impact of our attacking method on the AES-candidate RC6, whose design was based on RC5.

1 Introduction

The iterated block cipher RC5 was introduced by Rivest in [Riv95]. It has a variable number of rounds denoted with r and key size of b bytes. The design is word-oriented for word sizes $w = 32, 64$ and the block size is $2w$. The choice of parameters is usually denoted by RC5-w, RC5-w/r, or RC5-$w/r/b$. Currently RC5-32/16 is recommended to give sufficient resistance against linear and differential attacks [KY98].

RC5 has been analyzed intensively. For an overview we refer to the report by Kaliski and Yin [KY98]. Currently the best published attack can be found in [BK98]. There a chosen plaintext attack is described for which we summarize the complexities for different round versions of RC5[1] in the second column of Table 1. As this is a differential attack, it yields a known plaintext attack for a larger amount of known plaintexts [BS93]. We give the estimated required amount of known plaintexts in the third column of Table 1. The known plaintext attack however needs a storage capacity for all the required plaintexts, i.e., the attack can not be mounted in a way that the attacker obtains and analyzes the plaintexts one by one. We give the estimated required storage capacity of the known plaintext-version in the fourth column. For example, to mount this attack for 4 rounds one would need to store 2^{36} plaintexts with corresponding ciphertexts, which is about 1 GByte. In this paper we present an attack that

[1] Although the attack of [BK98] makes use of very sophisticated techniques, according to the authors the required amount of chosen plaintexts for the attack on 12 rounds might be reduced to 2^{38}.

requires a negligible storage capacity. We give the required amount of plaintexts in the fifth column of Table 1.

Table 1. Complexities (lg) of the attacks on RC5-32.

	Biryukov/Kushilevitz			Our attack	
Rounds	Chosen plaintexts	Known plaintexts	Storage	Known plaintexts	Storage
4	7	36	36	28	negligible
6	16	40.5	40.5	40	negligible
8	28	46.5	46.5	52	negligible
10	36	50.5	50.5	64	negligible
12	44	54.5	54.5	–	–

Our attack is a linear attack, whose high success rate is based on a large linear hull-effect [Nyb94]. To our knowledge it is the first time that this effect has significant consequences in the evaluation of the resistance of a cipher against linear attacks. Furthermore we use techniques closely related to multiple linear approximations to set up a practical attack.

Recently RC6 [RC6.1] has been submitted to NIST for the AES-Development Effort as a candidate to replace the DES as block cipher standard. The design of RC6 is based on RC5 and its public security analysis. Special adjustments were done to make RC6 resistant against the successful differential attacks on RC5. In this paper we also address the consequences for RC6 of our linear attack on RC5.

The remainder of this paper is organized as follows. In Sect. 2 we describe some techniques from linear cryptanalysis and show their merits and limitations when applied to RC5 in Sect. 3. In Sect. 4 we describe our attack on RC5 and we give experimental results on RC5-32 and RC5-64. We discuss the consequences for RC6 in Sect. 5 and conclude in Sect. 6.

2 Linear Cryptanalysis

Linear cryptanalysis was introduced and developed by Matsui in [Mat93,Mat94]. Additional advanced techniques, which are relevant for this paper can be found in [KR94,Nyb94,Vau96]. A basic linear attack makes use of a linear approximation between bits of the plaintext P, bits of the plaintext C and bits of the expanded key K. Such a linear approximation is a probabilistic relation that can be denoted as

$$\alpha \cdot P \oplus \beta \cdot K = \gamma \cdot C, \tag{1}$$

where α, β and γ are binary vectors and $\chi \cdot X = \bigoplus_i \chi_i X_i$ for $\chi = (\chi_0, \chi_1, \ldots)$. Instead of P or C one can use intermediate computational values that can be

computed from P or C under the assumption of a key value. If a linear approximation holds with probability $p = \frac{1}{2} + \delta$ with $\delta \neq 0$, a linear attack can be mounted which needs about $c|\delta|^{-2}$ plaintexts. The value of c depends on the attacking algorithm that is used. Here δ is called the deviation and $|\delta| = \epsilon$ is called the bias.

Since the key K is fixed, (1) can be transformed into (2) without changing the bias.

$$\alpha \cdot P = \gamma \cdot C, \tag{2}$$

We say that *for certain values $\tilde{P}$ and $\tilde{C}$, (2) behaves in the deviation direction* if $\alpha \cdot \tilde{P} \oplus \gamma \cdot \tilde{C} = b$ and $\alpha \cdot P \oplus \gamma \cdot C = b$ has a positive deviation.

A linear approximation for the whole cipher can be derived by 'chaining' linear approximations between intermediate values. If the probabilities of these approximations are *independent*, the value of the deviation of the derived approximation can be computed with Matsui's Piling Up Lemma [Mat93]. This states that the deviation δ of n chained approximations with deviations δ_i is given by

$$\delta = 2^{n-1} \prod_{i=1}^{n} \delta_i \,.$$

3 RC5 and Linear Cryptanalysis

RC5 is defined as follows. First $2r+2$ round keys $S_i \in \{0,1\}^w$, $i = 0, \ldots, 2r+1$, are derived from the user key.[2] If $(L_0, R_0) \in \{0,1\}^w \times \{0,1\}^w$ is the plaintext, then the ciphertext (L_{2r+1}, R_{2r+1}) is computed iteratively with:

$$L_1 = L_0 + S_0 \tag{3}$$
$$R_1 = R_0 + S_1 \tag{4}$$
$$U_i = L_i \oplus R_i \tag{5}$$
$$V_i = U_i \lll R_i \tag{6}$$
$$R_{i+1} = V_i + S_{i+1} \tag{7}$$
$$L_{i+1} = R_i \tag{8}$$

for $i = 1, \ldots, 2r$. Here $+$ denotes addition modulo 2^w and $x \lll y$ rotation of w-bit word x to the left over y mod w places. The computation of (L_{i+2}, R_{i+2}) from (L_i, R_i) with i odd is considered as one round of RC5. In Fig. 1 a graphical representation of one round is given.

3.1 Linear Approximations

We shall consider the following linear approximations for xor, data-dependent rotation and addition. We only look at approximations that consider one bit of

[2] As the key schedule of RC5 has no relevance for our analysis we refer to [Riv95] for a description. However for our experiments we have used the key schedule.

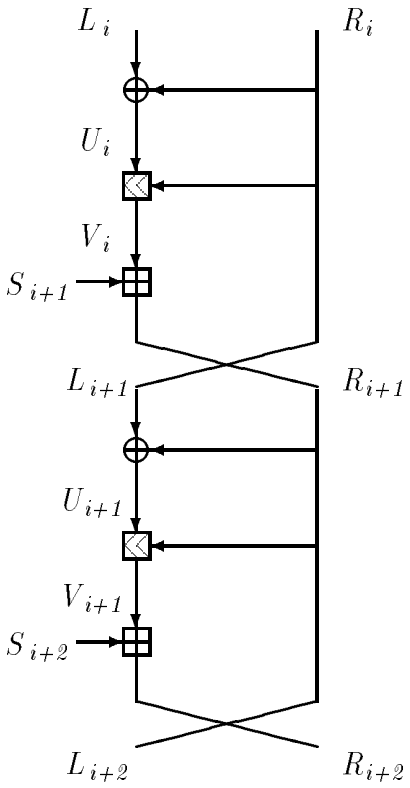

Fig. 1. One round of RC5.

each term of the equation. The binary vector that has a 1 on position i and is 0 everywhere else, will be denoted with e_i. Let $A = B \oplus C$. Then:

$$e_i \cdot A = e_i \cdot B \oplus e_i \cdot C, \quad \delta = 2^{-1} \tag{9}$$

for $i \in \{0, \ldots, w-1\}$. Let $D = E \ll F$. Then:

$$e_i \cdot D = e_j \cdot E \oplus e_k \cdot F \oplus e_k \cdot (i-j), \quad \delta = 2^{-\lg w - 1} \tag{10}$$

for $i, j \in \{0, \ldots, w-1\}$ and $k \in \{0, \ldots, \lg w - 1\}$. (Here we have abused the "$\cdot$"-notation slightly to denote the k-th bit in the binary representation of $i-j$.) If one is only interested in the bias of (10), one can leave out the term $e_k \cdot (i-j)$ or even $e_k \cdot F$, see for example [KY98]. We use (9) and (10) to pass the xor and rotation in RC5 as follows. Let $j, k \in \{0, \ldots, \lg w - 1\}$. Then

$$e_j \cdot U_i = e_j \cdot L_i \oplus e_j \cdot R_i, \quad \delta = 2^{-1} \tag{11}$$

$$e_k \cdot V_i = e_j \cdot U_i \oplus e_j \cdot R_i \oplus e_j \cdot (k-j), \quad \delta = 2^{-\lg w - 1}. \tag{12}$$

Chaining these two yields:

$$e_k \cdot V_i = e_j \cdot L_i \oplus e_j \cdot (k-j), \quad \delta = 2^{-\lg w - 1}. \tag{13}$$

Finally, let $G = H + S$, where S is fixed. Then:

$$e_0 \cdot G = e_0 \cdot H \oplus e_0 \cdot S, \quad \delta = 2^{-1} \tag{14}$$

$$e_i \cdot G = e_i \cdot H \oplus e_i \cdot S, \quad \delta = 2^{-1} - 2^{-i} S[i] \tag{15}$$

for $i \in \{1, \ldots, w-1\}$. Here $S[x] = S \bmod 2^{x+1}$, hence the x LSB's of S. Hence, depending on the key, the bias of (15) can vary between 0 and $\frac{1}{2}$. On the average it is $\frac{1}{4}$.

3.2 Key Dependency and Piling-Up

Because of the fact that (14) has a bigger bias than (15), 'traditionally' approximations on the LSB have been considered to be most useful for a linear attack (see [KY95,KY98]). Using Approximations (9), (10) and (14) one can derive the following iterative approximation for one round of RC5.

$$e_0 \cdot L_i \oplus e_0 \cdot S_{i+1} = e_0 \cdot L_{i+2} \ (i \geq 1), \quad \delta = 2^{-\lg w - 1} \tag{16}$$

This approximation can be chained to l rounds as follows.

$$e_0 \cdot L_i \oplus \bigoplus_{j=0}^{l-1} e_0 \cdot S_{i+1+2j} = e_0 \cdot L_{i+2l} \ (i \geq 1), \quad \delta = ? \tag{17}$$

According to the Piling Up Lemma (2) this approximation would have deviation $\delta = 2^{l-1} 2^{(-\lg w - 1)l} = 2^{-l \lg w - 1}$ if the chained approximations would be independent. This is however not the case.

To illustrate this consider (17) for $l = 2$. The deviation of this approximation depends on the $\lg w$ least significant bits of S_{i+2}. This can be seen as follows. The probability that (16), i.e., the approximation over the first round, holds depends on the value of $R_i \bmod w$. If $R_i \bmod w = 0$ it always holds, otherwise it holds with probability $\frac{1}{2}$. Hence, the deviation of (16) is computed under the assumption that every value of $R_i \bmod w$ is equally likely. If $R_i \bmod w = 0$ then the $\lg w$ least significant bits of L_{i+1} are known. Now consider the approximation for each possible value of R_{i+1} separately. If $R_{i+1} \bmod w \in \{0, \ldots, \lg w - 1, w - \lg w + 1, \ldots, w-1\}$ then, depending on the value of $S_{i+2} \bmod w$, part of the $\lg w$ least significant bits of R_{i+2} can be computed. It turns out that the values of R_{i+3} are not equally likely. Hence, the Piling Up Lemma cannot be applied.

To illustrate this effect we have computed the bias of the two round approximation for RC5-32 for every value of $S_{i+2} \bmod w$. These results are given in Table 2. It can be seen that the bias can vary significantly between $\approx 2^{-10}$ and $\approx 2^{-16}$. On the other hand the average value is $\frac{1}{32} \sum_{x=0}^{31} |\delta_x| \approx 2^{-11}$. Since $\delta_x \neq 0$ for all x, the average amount of expected plaintexts needed based on

Table 2 is given by $\frac{1}{32}\sum_{x=0}^{31}\delta_x^{-2} \approx 2^{22}$. Both are in accordance with the Piling Up Lemma.

Note further that for two values of $S_{i+2} \bmod w$ the deviation is negative. For those values this means that if one would mount a basic linear attack (Algorithm 1 in [Mat93]) on four rounds using (17) to find $S_0 \oplus S_2 \oplus S_4$ based on the Piling Up Lemma, most likely one would find $S_0 \oplus S_2 \oplus S_4 \oplus 1$, given enough texts.

We can conclude that although the deviation can vary significantly, the Piling Up Lemma gives a good estimate for the average bias for RC5. We will show that this estimate can be used to compute the expected average success rate of a linear attack that does not use the sign of the deviation, but only its absolute value; the bias.

Table 2. The deviation δ_x of Approximation (17) with $l = 2$ for RC5-32, depending on $x = S_{i+2} \bmod w$.

$x = S_{i+2} \bmod w$	$10^3\delta_x$	$\lg \lvert\delta_x\rvert$	$x = S_{i+2} \bmod w$	$10^3\delta_x$	$\lg \lvert\delta_x\rvert$
00	0.930786	-10.07	10	0.228882	-12.09
01	0.991821	- 9.98	11	0.656128	-10.57
02	0.473022	-11.05	12	0.015259	-16
03	0.564575	-10.79	13	-0.015259	-16
04	0.595093	-10.71	14	0.137329	-12.83
05	0.686646	-10.51	15	0.534058	-10.87
06	0.473022	-11.05	16	0.503540	-10.96
07	0.503540	-10.96	17	0.717163	-10.45
08	0.595093	-10.71	18	0.625610	-10.64
09	0.564575	-10.79	19	0.778198	-10.33
0a	0.289917	-11.75	1a	0.747681	-10.39
0b	0.381470	-11.36	1b	0.839233	-10.22
0c	0.411987	-11.25	1c	0.381470	-11.36
0d	0.289917	-11.75	1d	0.381470	-11.36
0e	0.106812	-13.19	1e	0.076294	-13.68
0f	0.228882	-12.09	1f	-0.045776	-14.42

3.3 Key Dependency and Linear Hulls

The concept (approximate) *linear hull* was introduced in [Nyb94]. We will use the term linear hull in the way it was used in [RC6.2]. A linear hull is the set of all chains of linear equations over (a part of) the cipher that produce the same linear equation. The existence of this effect for RC5 was first noticed in [RC6.2,RC6.3] where also some preliminary work in determining the linear hull effects for RC6 (and some simplified versions) can be found.

The following linear hull for an approximation of two rounds (i till $i+3$) of RC5 can be noticed. Let $j, l \in \{0, \ldots, w-1\}$. Using (9), (10), (14) and (15) for

j the following $\lg w$ approximations for two rounds can be derived[3].

$$e_j \cdot L_i \oplus e_j \cdot (k-j) \oplus e_k \cdot S_{i+1} = e_k \cdot L_{i+2}, \;\; \delta = \delta_{j,k} \tag{18}$$

for $k = 0, \ldots, \lg w - 1$. Likewise for the next two rounds and l also $\lg w$ approximations can be derived:

$$e_k \cdot L_{i+2} \oplus e_k \cdot (l-k) \oplus e_l \cdot S_{i+3} = e_l \cdot L_{i+4}, \;\; \delta = \delta_{k,l} \tag{19}$$

for $k = 0, \ldots, \lg w - 1$. Hence, one can chain $\lg w$ pairs of (18) and (19) to obtain the two round approximation

$$e_j \cdot L_i \oplus e_k \cdot S_{i+1} \oplus e_l \cdot S_{i+3} \oplus e_j \cdot (k-j) \oplus e_k \cdot (l-k) = e_l \cdot L_{i+4}, \; \delta = \delta_{j,k,l} \tag{20}$$

Neglecting the key-dependency of chaining and using the Piling Up Lemma one gets for this bias

$$\delta_{j,k,l} = \begin{cases} 2^{-2\lg w - 1} & k = 0, l = 0 \\ 2^{-2\lg w - 1}(1 - 2^{-k+1} S_{i+1}[k-1]) & k \neq 0, l = 0 \\ 2^{-2\lg w - 1}(1 - 2^{-l+1} S_{i+3}[l-1]) & k = 0, l \neq 0 \\ 2^{-2\lg w - 1}(1 - 2^{-k+1} S_{i+1}[k-1])(1 - 2^{-l+1} S_{i+3}[l-1]) & k \neq 0, l \neq 0 \end{cases} \tag{21}$$

We note two things about (20) and its deviation given in (21). Firstly it is clear from (21) that the deviation is key dependent, i.e., dependent on the $\lg w - 1$ LSB's of S_{i+1} and S_{i+3}. But we will show that because of the linear hull effect, this key dependency is negligible. Secondly, for each choice of the triple j, k, l the term $C_{j,k,l} = e_k \cdot S_{i+1} \oplus e_l \cdot S_{i+3} \oplus e_j \cdot (k-j) \oplus e_k \cdot (l-k)$ is constant, either 0 or 1. This constant actually determines the sign of the deviation of the following approximation, which can be derived from (20) by leaving out $C_{j,k,l}$.

$$e_j \cdot L_i = e_l \cdot L_{i+4}, \; \delta = \tilde{\delta}_{j,l} \, . \tag{22}$$

For the deviation the following holds:

$$\tilde{\delta}_{j,l} = \sum_{k \in V_{j,l}} \delta_{j,k,l} - \sum_{k \notin V_{j,l}} \delta_{j,k,l}, \tag{23}$$

where $V_{j,l} = \{k \in \{0, \ldots, \lg w - 1\} | C_{j,k,l} = 0\}$.

One can extend approximation (22) to hold for r subsequent rounds. In this way one gets the following approximation for r rounds.

$$e_j \cdot L_i = e_l \cdot L_{i+2r}, \; \delta = \tilde{\delta}_{j,l}, \tag{24}$$

where $i, j \in \{0, \ldots, w-1\}$. The deviation can be computed/approximated by considering the parts of the different chains, for the k-th round in the chain given by

$$e_{j_k} \cdot L_{i+2k} = e_{l_k} \cdot L_{i+2k+2}, \tag{25}$$

[3] Note that in Equation (18) and others $\delta_{j,k}$ is not the Kronecker delta, but a variable δ with indices j and k.

where $k \in \{0, \ldots, r-1\}$ and

$$j_k \begin{cases} = j & \text{if } k = 0 \\ \in \{0, \ldots, \lg w - 1\} & \text{if } k \neq 0 \end{cases}$$

$$l_k \begin{cases} \in \{0, \ldots, \lg w - 1\} & \text{if } k \neq r-1 \\ = l & \text{if } k = r-1 \end{cases}$$

and

$$j_k = l_{k-1} \text{ for } k \in \{1, \ldots, r-1\}.$$

In this way it can be seen that Equation (24) is a linear hull that consists of $(\lg w)^{r-1}$ different chains, namely for all of the choices of j_k, l_k. When we take the average bias of $\frac{1}{4}$ for the approximation of addition on non-LSB's, we get for the bias of (24):

$$|\tilde{\delta}_{j,l}| = \begin{cases} c(r) & \text{if } l = 0 \\ \frac{1}{2}c(r) & \text{if } l \neq 0 \end{cases} \tag{26}$$

where $c(r)$ can be estimated as

$$\begin{aligned} c(r) &\approx \gamma(\sum_{k=0}^{r-1} \binom{r-1}{k} (\lg w - 1)^k 2^{(-\lg w - 1)r + r - 1 - k}) \\ &= \gamma(2^{(-\lg w - 1)r} \sum_{k=0}^{r-1} \binom{r-1}{k} (\lg w - 1)^k 2^{r-1-k}) \\ &= \gamma(2^{(-\lg w - 1)r} (\lg w - 1 + 2)^{r-1}) \\ &= \gamma(\frac{1}{2w} \left(\frac{\lg w + 1}{2w}\right)^{r-1}) \end{aligned} \tag{27}$$

where γ is a factor that accounts for the effect of different chains that cancel each others contribution. The increasement factor $c^+(r)$ of the bias is expressed as

$$c(r+1) = c^+(r) \cdot c(r), \text{ for } r = 2, \ldots \tag{28}$$

Hence, if a linear attack can be mounted on RC5 with r rounds using x known plaintexts, then this attack will have the same success probability if adapted to RC5 with $(r+1)$ rounds if $(c^+(r))^{-2}x$ known plaintexts are available. From (27) and (28) follows $c^+ = \frac{\lg w + 1}{2w}$. Now, for RC5-32 we have $c^+(r) \approx 2^{-3.4}$ and for RC5-64 we have $c^+(r) \approx 2^{-4.2}$. In Sect. 4.4 we will show that the attacks that we have implemented for RC5-32 and RC5-64 approximately behave according to the previously derived approximate expectations. Hence, we can neglect the key dependency in (24): the bias given according to (26) is a sufficient practical estimate.

4 The Attack on RC5

In this section we derive a linear attack on RC5 and present an overview of the experimental results. In Sect. 4.1 we give the linear approximation and the

method that is used to guess key bits. In Sect. 4.2 we specify this method. In Sect. 4.3 we describe the search algorithm. Finally, in Sect. 4.4 experimental results of the implemented attack are given.

4.1 The Linear Approximations

We have derived and implemented a linear attack that uses approximation (24). Since this approximation does not involve any key bits, a basic linear attack is not possible. In particular for an r-round attack we will use (24) with $i = 0$ and $k = r$. The first addition, adding S_0, will be passed with an approximation on the LSB, since this has bias $\frac{1}{2}$. Therefore we take $j = 0$ in (24). We also choose $l = 0$ in (24) because then also the last key-addition in the whole approximation will be passed with bias $\frac{1}{2}$. It might be possible to further improve the attack by (also) using approximations with other values for j and l, but we have not found a method to do this.

To attack r rounds of RC5 we use the fact that each linear path that is part of the hull given by (24) is a chain of r 1-round approximations of the form (25). We will split the approximation into two parts. The first contains the approximation for the first key addition and the first round. This gives us the following $\lg w$ approximations. Each is the first part of a set of linear approximations that is contained in the linear hull of (24).

$$e_0 \cdot L_0 = e_k \cdot L_3 \text{ for } k = 0, \ldots, \lg w - 1. \tag{29}$$

The remainder of the whole approximation can be specified by the following $\lg w$ approximations, each beginning with a different bit of L_3:

$$e_k \cdot L_3 = e_0 \cdot L_{2r+1} \text{ for } k = 0, \ldots, \lg w - 1\,. \tag{30}$$

When for a certain plaintext encryption the intermediate value $R_1 \bmod w = k$, hence an element of $\{0, 1, \ldots, \lg w - 1\}$, then (29) behaves in the deviation direction. Hence, if also (30) behaves in the deviation direction then the whole approximation behaves in that direction. On the other hand, if $R_1 \bmod w \notin \{0, 1, \ldots, \lg w - 1\}$ then the probability that the whole approximation behaves in the direction of the deviation is much lower.

In the attack we want to check for every text if one of the approximations that correspond to (30) was followed. Since we have no information about any intermediate values we do not have a criterion that always holds. Instead we will derive a function that is expected to give higher values when one of the approximations was followed. Hence this function will have higher values for encryptions where $R_1 \bmod w \in \{0, 1, \ldots, \lg w - 1\}$ than for other R_1 values. Because $R_1 \bmod w = (R_0 - S_1) \bmod w$ and R_0 is known we can guess $S_1 \bmod w$ from this. We call this function the *non-uniformity function*. In the next section we will describe it.

We note here that the use of such a function fits in the frameworks of statistical cryptanalysis as described by Vaudenay in [Vau96] or partitioning cryptanalysis as described by Harpes and Massey in [HM97].

4.2 A Non-Uniformity Function

The non-uniformity function ν computes non-uniformity values for a given set of corresponding plaintext/ciphertext pairs. This set is divided into w subsets, each set contains plaintexts with the same $R_0 \bmod w$-value. For each set a non-uniformity value can be computed.

We look at the last round of the encryption. Suppose that one approximation of the linear hull corresponding to (30) is followed up to the last round. Say that in this particular case $e_k \cdot L_{2r-1}$ was biased for some $k \in \{0, \ldots, \lg w - 1\}$. Then following the approximation to the end would mean that $R_{2r-1} \bmod w = (w - k) \bmod w$ with a higher probability than other values. As seen in Sect. 3.2 depending on the subkeys also the other possible values of $R_{2r-1} \bmod w$ might not be uniformly distributed. But in any case certain values of $R_{2r-1} \bmod w$ will have a higher probability when (30) behaves in the deviation direction then when it does not behave in that way.

If we would know the value of $R_{2r-1} \bmod w$ it would be possible to compute $\lg w$ bits of S_{2r+1} or two possible values for those bits from the ciphertext with:

$$S_{2r+1} = R_{2r+1} - (R_{2r-1} \oplus L_{2r+1}) \lll L_{2r+1}, \tag{31}$$

since the values of R_{2r+1} and L_{2r+1} are known from the ciphertext. We will use $S\langle n \rangle$ to denote the $\lg w$-bit string, given by the bits $(n + \lg w - 1) \bmod w, \ldots, (n+1) \bmod w, n$ of S. The value of $L_{2r+1} \bmod w$ determines for which $\lg w$ bits of S_{2r+1} information can be computed, namely $S\langle L_{2r+1} \bmod w \rangle$. If $L_{2r+1} \bmod w = 0$ then the $\lg w$ LSB's can be computed. When $L_{2r+1} \bmod w \neq 0$ we can compute two values for $S\langle L_{2r+1} \bmod w \rangle$. The carry bit of the addition in (31) determines which one is the correct one.

In the attack we do not know the value of $R_{2r-1} \bmod w$ or even which value would be the most probable. Instead of trying to compute $\lg w$ bits of S_{2r+1} we make a similar computation for a value which we call S'. It is computed by taking $R_{2r-1} \bmod w = 0$ in (31), i.e.,

$$S' = R_{2r+1} - (L_{2r+1} \lll L_{2r+1}). \tag{32}$$

Due to the non-uniform distribution of $R_{2r-1} \bmod w$ it is expected that the distribution of $S'\langle \cdot \rangle$-values will be more non-uniform for encryptions where the approximation was followed than for others.

Hence, in the attack we use a counter array $A(i, j, k)$ for $i, j, k = 0, \ldots, w-1$, where each i corresponds to a possible value of $R_0 \bmod w$, each j to a possible value of $L_{2r+1} \bmod w$ and each k to a possible value of $S'\langle L_{2r+1} \bmod w \rangle$. For each text we check if the approximation holds. If it holds, we change the counter array as follows. If $L_{2r+1} \bmod w = 0$, we increase $A(R_0 \bmod w, L_{2r+1} \bmod w, v)$ by 2, where v is the suggested $S'\langle L_{2r+1} \bmod w \rangle$-value. If $L_{2r+1} \bmod w \neq 0$, we increase $A(R_0 \bmod w, L_{2r+1} \bmod w, v_0)$ and $A(R_0 \bmod w, L_{2r+1} \bmod w, v_1)$ by 1, where v_0 and v_1 are the suggested values. If the approximation does not hold, we decrease the specific array entries accordingly.

Each $(R_0 \bmod w,\ L_{2r+1} \bmod w)$-combination gives a distribution of $S'\langle \cdot \rangle$-values. From Sect. 4.1 we know that the distributions corresponding to the values

$R_1 \bmod w \in \{0, \ldots, \lg w - 1\}$ will be the most non-uniform. To measure the non-uniformity we check for all w bits of our S' based on the $S'\langle\cdot\rangle$-values how many times 0 is suggested and how many times 1 and take the difference of these amounts. For each $R_0 \bmod w$ we take the sum over all possible $L_{2r+1} \bmod w$ of the absolute values of these differences. In this way the *non-uniformity function* $\nu : \{0, \ldots, w-1\} \to \mathbf{N}$ is defined:

$$\nu(r_0) = \sum_{l_{2r+1}=0}^{w-1} \sum_{x=0}^{\lg w - 1} \Big| \sum_{v : e_x \cdot v = 0} A(r_0, l_{2r+1} + x, v) - \sum_{v : e_x \cdot v = 1} A(r_0, l_{2r+1} + x, v) \Big|, \tag{33}$$

where all indices of A are taken mod$w$. We call the w values that are derived in this way the *non-uniformity values*. We expect that the sum of $\lg w$ non-uniformity values will be maximal for the values corresponding to texts with $R_1 \bmod w \in \{0, \ldots, \lg w - 1\}$. The final step of the algorithm guesses the value of $S_1 \bmod w$ accordingly.

The observant reader will have noticed that according to the above description it is not necessary to have counters for the $S'\langle\cdot\rangle$-values. Instead of this one could use counters corresponding to the bits of S' and change these accordingly. The description above is used to emphasize that with the $S'\langle\cdot\rangle$-distribution also other non-uniformity measurements could be used. For example, one could look at all possible values for two subsequent bits of S'. Also it is probably possible to derive information about the actual value of S_{2r+1} from the $S'\langle\cdot\rangle$-distribution. However, this falls outside the scope of this paper.

4.3 The Algorithm

1. Acquire n known plaintext/ciphertext-pairs $(P_0, C_0), \ldots, (P_{n-1}, C_{n-1})$.
2. Initialize a counter array[4] $A(i, j, k) := 0$ for $i, j, k = 0, \ldots, w - 1$.
3. For each plaintext/ciphertext-pair do:
   If $L_{2r+1} \bmod w = 0$ then
   (a) Compute $S'\langle 0\rangle$-guess v.
   (b) If $e_0 \cdot L_0 = e_0 \cdot L_{2r+1}$ then $A(R_0, L_{2r+1}, v) := A(R_0, L_{2r+1}, v) + 2$.
   If $e_0 \cdot L_0 = e_0 \cdot L_{2r+1} \oplus 1$ then $A(R_0, L_{2r+1}, v) := A(R_0, L_{2r+1}, v) - 2$.
   If $L_{2r+1} \bmod w \neq 0$ then
   (a) Compute $S'\langle L_{2r+1} \bmod w\rangle$-guesses v_0 and v_1.
   (b) If $e_0 \cdot L_0 = e_0 \cdot L_{2r+1}$ then $A(R_0, L_{2r+1}, v_0) := A(R_0, L_{2r+1}, v_0) + 1$ and $A(R_0, L_{2r+1}, v_1) := A(R_0, L_{2r+1}, v_1) + 1$.
   If $e_0 \cdot L_0 \neq e_0 \cdot L_{2r+1}$ then $A(R_0, L_{2r+1}, v_0) := A(R_0, L_{2r+1}, v_0) - 1$ and $A(R_0, L_{2r+1}, v_1) := A(R_0, L_{2r+1}, v_1) - 1$.
4. Compute w non-uniformity values $\nu(i)$ according to (33), where i corresponds with a value of $R_0 \bmod w$.
5. Find the value $x \in \{0, \ldots, w-1\}$ for which $\sum_{i=0}^{\lg w - 1} |\nu((x+i) \bmod w)|$ is maximal.
6. Guess $S_1 \bmod w = w - x$.

[4] For clarity in the following description of the algorithm we have left out mod w when referring to indices of A.

4.4 The Results

We have implemented the attack on RC5-32 and RC5-64. The results are given in Table 3 and Table 4. As can be seen in the tables, we have done tests for up to 5 rounds of RC5-32 and up to 4 rounds of RC5-64. For each number of rounds tests were performed for several amounts of plaintexts. To give an indication of the practical aspects of the experiments: with our RC5-implementation carrying out the attack on the 5-round version for 10 keys with 2^{32} plaintexts took about 21 hours on a 333 MHz Pentium. To our knowledge these are the first experimentally executed known plaintext attacks on reduced versions of RC5, which require a negligible storage.

As stated at the end of Sect. 3.3, we would expect that an attack on r rounds of RC5 with x known plaintexts should have the same success probability as an attack on $r+1$ rounds with $(c^+)^{-2}x$ texts. For RC5-32 it holds that $(c^+)^{-2} \approx 2^{6.8}$, for RC5-64 $(c^+)^{-2} \approx 2^{8.4}$. It can be seen from the tables that the results are better than expected, i.e., the factor to attack an extra round is $\approx 2^6$ for RC5-32 and $\approx 2^8$ for RC5-64. We conjecture that the reason for this is that the value of $R_{i+2} \bmod w$ depends significantly on the value of $R_i \bmod w$. We are still researching this problem, but we give some preliminary evidence in the next section, where we discuss the consequences for RC6.

Based on the experimental results and the theoretical estimation of the bias of the linear approximation we can estimate the complexity of the attack on more than 4 or 5 rounds (cf. Table 5). It follows that an attack can be mounted on RC5-32 with 10 rounds that has a success probability of 45% if 2^{62} plaintexts are available. An attack on RC5-64 with 15 rounds has a success probability of 90% if 2^{123} plaintexts are available.

5 Consequences for RC6

The block cipher RC6 [RC6.1] has been submitted to NIST as an AES-candidate. Its design was based on RC5 and the security evaluation of RC5. To meet the block size requirement of 128 bits and to keep a 32-bit processor oriented design for this block size, RC6 was designed as two RC5-32's (with some changes) in parallel that interact[5]. Hence, cryptanalysis of RC5 can mostly be adapted for analysis of RC6.

However, the RC5-structure used in RC6 differs from the original version. One of the most important differences is the following. The amount of rotation in RC5-32 was determined by taking the 5 LSB's of a 32-bit word. In RC6, the 5 bits that determine the amount of rotation depend on all 32 bits of a 32-bit word.

In the first place these changes to RC5 were made to preclude the successful differential attacks on RC5 [BK98,RC6.1]. These attacks make use of the fact

[5] Actually, the design of RC6 also is word oriented and the blocksize is $4w$, where w is the word size. We only discuss the 128-block size version, as it is the main object for the AES standardization process.

Table 3. Experimental results of the attack on RC5-32.

Rounds	Known plaintexts	Success rate
2	2^{13}	28/100
	2^{14}	46/100
	2^{15}	89/100
	2^{16}	92/100
3	2^{17}	7/100
	2^{18}	15/100
	2^{19}	28/100
	2^{20}	49/100
	2^{21}	69/100
	2^{22}	81/100
4	2^{24}	7/100
	2^{25}	26/100
	2^{26}	44/100
	2^{27}	77/100
	2^{28}	82/100
5	2^{32}	4/10
	2^{33}	7/10
	2^{34}	9/10

Table 4. Experimental results of the attack on RC5-64.

Rounds	Known plaintexts	Success rate
2	2^{17}	39/100
	2^{18}	82/100
	2^{19}	96/100
3	2^{25}	28/50
	2^{26}	40/50
	2^{27}	47/50
4	2^{34}	9/10

Table 5. Expected number of plaintexts needed for a known plaintext attack on $r(\geq 2)$ rounds of RC5-32 or RC5-64.

Success probability:	45%	90%
RC5-32	2^{2+6r}	2^{4+6r}
RC5-64	2^{1+8r}	2^{3+8r}

that only five LSB's determine the rotations. However, these changes also provide increased resistance to the attack method described in this paper.

To illustrate this we look at a transition version between RC5 and RC6. In the definition of RC5, replace (5) and (6) with

$$T_i = (R_i(2R_i + 1)) \lll 5 \tag{34}$$

$$U_i = L_i \oplus T_i \tag{35}$$

$$V_i = U_i \lll T_i \tag{36}$$

For our theoretical analysis concerning the linear hull effect, this change makes little difference. One can still use the same linear approximations. However, the first round trick and last round trick we have used now become more complicated. To compute T_1 mod 32, one has to guess all bits of S_1 and the construction of a non-uniformity function is not obvious.

We have done tests on the above described intermediate version with the last and first round replaced with a normal RC5-round. Then the first and last round trick are straightforward. We have implemented the attack for 3 and 4 rounds of the cipher and results indicate that the increase in the number of necessary plaintexts for the same success probability was more in accordance with the theoretical results for RC5. Hence, the application of the extra function to determine the rotation amounts causes these values to be more independent.

We conclude that our attack does not give an obvious possibility to mount a realistic attack on RC6. Currently we are working on a precise evaluation of its resistance against this attack method.

6 Conclusions

In this paper we have evaluated the resistance of RC5 against linear attacks. We have taken into account the applicability of the Piling Up Lemma and the consequences of linear hull effects, both in combination with possible key dependency. This resulted in estimates for the complexity to mount a linear attack.

Furthermore we have described an attack that exploits the linear hull effect that we described and implemented it on reduced versions of RC5-32 and RC5-64. In this way we estimate that our attack can theoretically break RC5-32 with 10 rounds and RC5-64 with 15 rounds. In comparison with previous attacks on RC5, our attack needs negligible storage capacity, i.e., it could be practically implemented.

The attack method has no serious consequences for the security of RC6. Apparently the precautions that the designers made to make RC6 more resistant against differential attacks, also made RC6 more resistant against more sophisticated linear attack methods.

Acknowledgments

We would like to acknowledge Antoon Bosselaers for providing a very efficient implementation of RC5, with which most of the experiments could be done in a

reasonable time. Furthermore we would like to thank Vincent Rijmen and Matt Robshaw for interesting discussions and helpful suggestions. We would also like to thank Ali Selçuk for many interesting discussions, as well as for pointing out the derivation of (27).

References

[BK98] A. Biryukov, E.Kushilevitz, "Improved Cryptanalysis of RC5," *Proc. Eurocrypt'98, LNCS 1403*, Springer-Verlag, 1998, pp. 85–99.

[BS93] E. Biham, A. Shamir, *Differential Cryptanalysis of the Data Encryption Standard*, Springer-Verlag, 1993.

[HM97] C. Harpes, J.L. Massey, "Partitioning Cryptanalysis," *Fast Software Encryption, LNCS 1267*, Springer-Verlag, 1997, pp. 13–27.

[KM96] L.R. Knudsen, W. Meier, "Improved Differential Attacks on RC5," *Proc. Crypto'96, LNCS 1109*, Springer-Verlag, 1996, pp. 216–228.

[KR94] B.S. Kaliski, M.J.B. Robshaw, "Linear Cryptanalysis Using Multiple Approximations," *Proc. Eurocrypt'94, LNCS 950*, Springer-Verlag, 1995, pp. 26–39.

[KY95] B.S. Kaliski, Y.L. Yin, "On Differential and Linear Cryptanalysis of the RC5 Encryption Algorithm," *Proc. Crypto'95, LNCS 963*, Springer-Verlag, 1995, pp. 171–184.

[KY98] B.S. Kaliski, Y.L. Yin, *"On the Security of the RC5 Encryption Algorithm,"* RSA Laboratories Technical Report TR-602, Version 1.0, September 1998, available via `http://www.rsa.com/rsalabs/aes`.

[Mat93] M. Matsui, "Linear cryptanalysis method for DES cipher," *Proc. Eurocrypt'93, LNCS 765*, Springer-Verlag, 1994, pp. 386–397.

[Mat94] M. Matsui, "The first experimental cryptanalysis of the Data Encryption Standard," *Proc. Crypto'94, LNCS 839*, Springer-Verlag, 1994, pp. 1–11.

[Nyb94] K. Nyberg, "Linear Approximations of Block Ciphers," *Proc. Eurocrypt'94, LNCS 950*, Springer-Verlag, 1995, pp. 439–444.

[Riv95] R.L. Rivest, "The RC5 Encryption Algorithm," *Fast Software Encryption, LNCS 1008*, Springer-Verlag, 1995, pp. 86–96.

[RC6.1] R.L. Rivest, M.J.B. Robshaw, R. Sidney, Y.L. Yin, *"The RC6 Block Cipher. v1.1,"* AES Proposal, 1998, available via `http://www.rsa.com/rsalabs/aes`.

[RC6.2] S. Contini, R.L. Rivest, M.J.B. Robshaw, Y.L. Yin, *"The Security of the RC6 Block Cipher. v1.0,"* 1998, available via `http://www.rsa.com/rsalabs/aes`.

[RC6.3] S. Contini, R.L. Rivest, M.J.B. Robshaw, Y.L. Yin, *"Linear Hulls and RC6 (DRAFT),"* September, 1998.

[Sel98] A.A. Selçuk, "New Results in Linear Cryptanalysis of RC5," *Fast Software Encryption, LNCS 1372*, Springer-Verlag, 1998, pp. 1–16.

[Vau96] S. Vaudenay, "An Experiment on DES - Statistical Cryptanalysis," *Proc. 3rd ACM Conference on Computer Security*, ACM Press, 1996, pp. 139–147.