# A Statistical Attack on RC6

Henri Gilbert[1], Helena Handschuh[2], Antoine Joux[3], and Serge Vaudenay[4]

[1] France Telecom
[2] Gemplus
[3] SCSSI
[4] Ecole Normale Supérieure – CNRS
Contact Helena.Handschuh@gemplus.com

**Abstract.** This paper details the attack on RC6 which was announced in a report published in the proceedings of the second AES candidate conference (March 1999). Based on an observation on the RC6 statistics, we show how to distinguish RC6 from a random permutation and to recover the secret extended key for a fair number of rounds.

## 1 Introduction

RC6 is one of the 15 candidate algorithms that were presented at the first Advanced Encryption Standard candidate conference in August 1998. It was submitted by RSA laboratories [9] and has been selected as one of the five finalists for the second round of the AES contest organized by NIST [1].

In this paper, we first show the existence of a statistical weakness in RC6 which allows to mount a distinguisher attack on a reduced number of rounds. This means that given a certain number of plaintext-ciphertext pairs, an attacker is able to distinguish RC6 from a random permutation. A distinguisher for the $r$-round version of a cipher may often be converted into a key-recovery attack on $r+1$ or even more rounds. Matsui's linear cryptanalysis of DES provides a typical example of such a situation [7]. This also holds for RC6 : we show that we can gain one round as compared with our distinguisher to recover the extended keys of RC6 reduced to 14 rounds (or equivalently 15 RC6 inner rounds).

The paper is organised as follows : in the next Section we give the outlines of RC6 and in Section 3 we present the probabilistic event which leaks information. In Section 4 we explicitely construct the distinguisher and in Section 5 we adapt the latter to recover the extended secret key. Finally, we shortly discuss the case of RC5 and conclude.

## 2 RC6 Outlines

RC6 is characterized by three parameters $(w, r, b)$. It is dedicated to $w$-bit microprocessors and encrypts $4w$-bit blocks by using four registers. (We assume that $w$

B. Schneier (Ed.): FSE 2000, LNCS 1978, pp. 64–74, 2001.

is an integral power of 2.) It has $r$ rounds and uses a $b$-byte secret key. The nominal parameters for AES are $(32, 20, 16)$, $(32, 20, 24)$ and $(32, 20, 32)$, respectively for a 128, 196 and 256-bit user key. There is a key scheduling algorithm which extends the original $b$-byte key into an $2r + 4$-word array $S = (S_0, \ldots, S_{2r+3})$. In this paper, we will only use the $w$ and $r$ parameters, so we consider that the encryption is performed by using an arbitrary $2r + 4$-word array $S$ which plays the role of the secret key.

The encryption is performed by using four registers $A, B, C, D$. The algorithm is described by the following pseudo-code.

```
Input: $(A, B, C, D)$
    1. $B \leftarrow B + S_0$, $D \leftarrow D + S_1$
    2. for $i = 1$ to $r$ do
           $A \leftarrow ((A \oplus f(B)) \lll f(D)) + S_{2i}$
           $C \leftarrow ((C \oplus f(D)) \lll f(B)) + S_{2i+1}$
           $(A, B, C, D) \leftarrow (B, C, D, A)$
    3. $A \leftarrow A + S_{2r+2}$, $C \leftarrow C + S_{2r+3}$
Output: $(A, B, C, D)$
```

Here the $f$ function plays the role of a pseudo-random generator defined by

$$f(x) = g(x) \bmod 2^w \lll log_2 w = x(2x + 1) \bmod 2^w \lll log_2 w \ .$$

A picture of the RC6 encryption algorithm is given hereafter.

RC6 is very similar to RC5 in that it uses only simple operations such as binary addition, exclusive or and circular rotations. In addition, RC6 performs a simple modular multiplication.

Our results show that a reduced number of rounds of RC6 may be distinguished from a random permutation, which in turn enables an attacker to recover the secret keys of RC6 with one more round. This analysis also partly transposes to RC5. We would like to mention that an outline of our attack was introduced for the first time at the second AES conference in Rome in March 1999 [2], and that another paper dealing with the same kind of RC6 statistics [6] appears in these proceedings. However, the work reported in [6] and the work reported here are quite independent, both approaches for handling the RC6 statistics differ to some extent, and we feel it is important to present the attack announced in [2] in details here. Interestingly, both papers show that we can distinguish RC6 from a random permutation in polynomial time for a fair number of rounds, although it has been made clear [4,8] that the RC6 frame provides a pseudorandom permutation after five rounds once the data-dependent rotations are removed.

## 3 A Probabilistic Event on RC6 Encryption

For $1 \leq i \leq r$ and $0 \leq j < 4$, we let $R_{i,j}(S, a, b, c, d)$ denote the value of the register with index $j$ (considering that index 0 is for $A$, index 1 is for $B$, ...) after

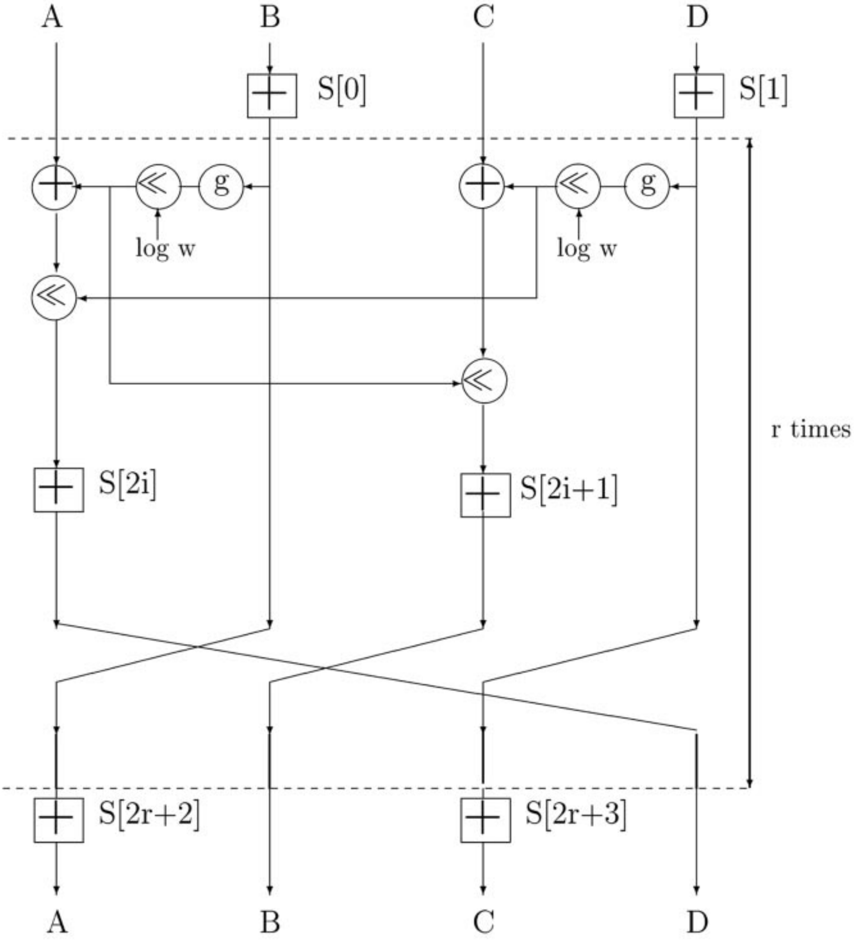

**Fig. 1.** Encryption with RC6-$w/r/b$ where $g(x) = x \times (2x + 1)$.

the $i$th round when the input of the encryption is $(a, b, c, d)$ and the key is $S$. We can extend the above notation to $i = 0$ letting $R_{0,j}(S, a, b, c, d)$ denote the input words to the first round. We will omit $(S, a, b, c, d)$ in most cases. In the sequel we implicitely assume that the $j$ index is taken modulo 4. We start with the following simple fact.

**Lemma 1.** *For any $i$ and any $(S, a, b, c, d)$, we have*

$$\left.\begin{array}{l} f(R_{i-1,1}) \equiv 0 \pmod{w} \\ f(R_{i-1,3}) \equiv 0 \pmod{w} \end{array}\right\} \Longrightarrow \left\{\begin{array}{l} R_{i,3} - R_{i-1,0} \equiv S_{2i} \pmod{w} \\ R_{i,1} - R_{i-1,2} \equiv S_{2i+1} \pmod{w} \end{array}\right.$$

*and in addition, $R_{i,0} = R_{i-1,1}$ and $R_{i,2} = R_{i-1,3}$.*

This comes from the fact that if the mod $w$ part of $f(B)$ and $f(D)$ are both zero in the $i$th round, then nothing is XORed onto the mod $w$ part of $A$ and $C$, and none are rotated.

This fact extends into the following

**Lemma 2.** *If we have* $f(R_{i-1,1}) \equiv f(R_{i-1,3}) \equiv 0 \pmod{w}$ *for* $i = k, k+2, \ldots, k+2\ell$*, then* $R_{k+2\ell,-2\ell-1} - R_{k-1,0} \bmod w$ *and* $R_{k+2\ell,1-2\ell} - R_{k-1,2} \bmod w$ *are constants which only depend on* $S$*.*

Assuming that the outputs of $f$ mod $w$ behave like random numbers, this event holds with probability $w^{-2\ell}$. We thus have the following heuristic result which has been confirmed by statistical experiments for $w = 32$ and small values of $r$.

**Theorem 1.** *Under heuristic assumptions, there exists some functions* $c_1(S)$ *and* $c_2(S)$ *such that for random* $(R_{0,0}, \ldots, R_{0,3})$ *and a random* $S$ *we have*

$$\Pr\begin{bmatrix} R_{r,1-r}(S) - R_{0,1}(S) \bmod w = c_1(S) \\ R_{r,3-r}(S) - R_{0,3}(S) \bmod w = c_2(S) \end{bmatrix} \approx w^{-2}\left(1 + w^{-2\lfloor \frac{r}{2} \rfloor}\right).$$

## 4 On Distinguishing RC6 from a Random Permutation

We can construct a distinguisher between RC6 and a random permutation by using the above theorem through a known plaintext attack.

1. The distinguisher first gets $n$ random samples $(x_i, \mathrm{Enc}(x_i))$ where
$$x_i = (x_{i,0}, x_{i,1}, x_{i,2}, x_{i,3})$$
and
$$\mathrm{Enc}_i = (y_{i,0}, y_{i,1}, y_{i,2}, y_{i,3}).$$
2. Then it hashes the samples onto
$$h_i = (y_{i,1-r} - x_{i,1} \bmod w, y_{i,3-r} - x_{i,3} \bmod w).$$
3. It then creates $w^2$ counters which correspond to possible $h_i$ values and counts the number $n_{(u,v)}$ of $i$ indices such that $h_i = (u, v)$.
4. If the maximum of all $n_{(u,v)}$ is greater than a given threshold $t$, output 1, otherwise, output 0.

We let $\epsilon \approx w^{-2\lfloor \frac{r}{2} \rfloor}$ denote the probability that the event of Theorem 1 occurs for RC6. We need to compute the advantage in terms of $n, t, \epsilon$ of this attack for distinguishing RC6 from a random permutation.

Let us choose $t = n.w^{-2} + \delta$. ($nw^{-2}$ is the expected value of one counter for random hashes so $\delta$ measures the deviation from the ideal expected case.)

The probability $p$ that the distinguisher outputs 1 for RC6 is greater than the probability that the counter which corresponds to the constant values in

Theorem 1 is greater than $t$. When $n$ is large, this counter tends towards a normal law with expected value $n(w^{-2}(1-\epsilon)+\epsilon)$ (which we approximate by $nw^{-2}+n\epsilon$) and variance approximately $nw^{-2}$. We let

$$\varphi(x)=\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x}e^{-\frac{t^2}{2}}dt.$$

We have

$$p\approx\varphi\left(-\frac{t-nw^{-2}-n\epsilon}{\sqrt{n}w^{-1}}\right)+\left(1-\varphi\left(\frac{t-nw^{-2}}{\sqrt{n}w^{-1}}\right)^{w^2-1}\right)$$

which is

$$p\geq\varphi\left(-\frac{t-nw^{-2}-n\epsilon}{\sqrt{n}w^{-1}}\right).$$

This means

$$p\geq\varphi\left(-\delta wn^{-\frac{1}{2}}+w\epsilon\sqrt{n}\right). \tag{1}$$

Now the probability $p^*$ that the distinguisher outputs 1 for a random permutation is less that $w^2$ times the probability that one given counter is greater then $t$. This counter tends to behave like a normal law with expected value $nw^{-2}$ and variance $nw^{-2}$. We thus have

$$p^*\leq w^2\varphi\left(-\frac{t-nw^{-2}}{\sqrt{n}w^{-1}}\right)$$

which means

$$p^*\leq w^2\varphi\left(-\delta wn^{-\frac{1}{2}}\right). \tag{2}$$

Therefore the advantage for distinguishing RC6 from a random permutation is

$$\mathrm{Adv}\geq\varphi\left(-\delta wn^{-\frac{1}{2}}+w\epsilon\sqrt{n}\right)-w^2\varphi\left(-\delta wn^{-\frac{1}{2}}\right).$$

If we derive this function with respect to $\delta$, we obtain the maximum when the derivative is equal to zero. The choice of $\delta$ which maximizes this right hand term is

$$\delta=\frac{2\log w}{\epsilon w^2}+\frac{\epsilon n}{2}$$

for which

$$\mathrm{Adv}\geq\varphi\left(-\frac{2\log w}{\epsilon w\sqrt{n}}+\frac{\epsilon w\sqrt{n}}{2}\right)-w^2\varphi\left(-\frac{2\log w}{\epsilon w\sqrt{n}}-\frac{\epsilon w\sqrt{n}}{2}\right).$$

This analysis leads to the following result.

**Theorem 2.** *Let* $\alpha = \frac{\epsilon w \sqrt{n}}{2\sqrt{\log w}}$*. Under heuristic assumptions, the above distinguisher, when used with*

$$t = \frac{n}{w^2} + \frac{2\log w}{\epsilon w^2} + \frac{\epsilon n}{2} \quad \text{and} \quad n \geq 4\alpha^2 w^{4\lfloor \frac{r}{2} \rfloor - 2} \log w$$

*has an advantage greater than*

$$\mathrm{Adv} \geq \varphi\left(\sqrt{\log w}(-\alpha^{-1} + \alpha)\right) - w^2 \varphi\left(\sqrt{\log w}(-\alpha^{-1} - \alpha)\right).$$

Considering $\alpha = 5$ we have

$$\mathrm{Adv} \geq \varphi\left(\frac{24}{5}\sqrt{\log w}\right) - w^2 \varphi\left(-\frac{26}{5}\sqrt{\log w}\right)$$

with a complexity of

$$n \geq 100 w^{4\lfloor \frac{r}{2} \rfloor - 2} \log w.$$

We have to be concerned that the total number of samples cannot be greater than $2^{4w}$, which is the total number of possible plaintexts. Hence the above attack is significant for

$$r \leq 2\left\lfloor \frac{4w - 7 - \log_2 \log w}{4\log_2 w} + \frac{1}{2} \right\rfloor + 1 \ .$$

As an application, with the nominal choice $w = 32$ we obtain an advantage greater than $1 - 2^{-60}$ with a complexity of $n \approx 2^{20\lfloor \frac{r}{2} \rfloor - 2}$. Thus we can break up to $r = 13$ rounds (with $n = 2^{118}$).

# 5 On Recovering the Secret Key

## 5.1 A Simplified Approach for Recovering $S_0$ and $S_1$

Let us focus on nominal RC6 reduced to $r = 14$ rounds for a moment. Then a way of adapting the distinguisher to recover the whole secret key for RC6 reduced to 14 rounds is by a known plaintext attack which proceeds in the following way.

Suppose we black box encrypt a multiple $m$ of the $n$ plaintexts required by the previously described distinguisher on 13 rounds, thus obtaining $m$ $(x_i, y_i)$ plaintext-ciphertext pairs for the 14-round RC6. Let $\Delta A = A_{\mathrm{out}} - A_{\mathrm{in}} \pmod{w}$ where $A_{\mathrm{out}} = y_{i,-14}$ and $A_{\mathrm{in}} = x_{i,0}$ denote the input-output difference of the $\log_2 w$ least significant bits of the input word $A$ and similarly let $\Delta C = C_{\mathrm{out}} - C_{\mathrm{in}} \pmod{w}$ where $C_{\mathrm{out}} = y_{i,2-14}$ and $C_{\mathrm{in}} = x_{i,2}$ denote the difference modulo $w$ on input word $C$. For those $(x_i, y_i)$ pairs such that the $A$ and

$C$ input words are not rotated at the first round, $\Delta A$ and $\Delta C$ are equal, up to the unknown constants $S_2 \pmod w$ and $S_3 \pmod w$, to the $h_i$ differences considered in the 13-rounds distinguisher of Section 4.

The exhaustive trial of all the $S_0$ and $S_1$ keys, i.e. the computation for each $(S_0, S_1)$ key assumption, of the $(\Delta A, \Delta C)$ frequencies distribution on the subset of plaintext-ciphertext pairs such that no $A$ and $C$ rotations occur at the first round, followed by the 13-round distinguisher test of Section 4, can be performed in an efficient way which avoids processing each plaintext individually for each key assumption.

• First we generate a table of $2^{2w+2\log_2 w}$ (e.g. $2^{74}$ for nominal RC6) elements, where each entry is the frequency observed for the plaintext-ciphertext pairs according to the value of the $B$ and $D$ input words as well as the $\Delta A$ and $\Delta C$ input-output differences modulo $w$ (i.e. a potential value of the constant differences if all the rotations were zero as in our model).

• Now for approximately $2^{w-\log_2 w}$ (e.g. $2^{27}$) "good" $B$ values, we obtain that $f(B+S_0) \equiv 0 \pmod w$ in the first round. Therefore for each possible choice of the first subkey $S_0$, we may add together the frequencies of the $2^{w-\log_2 w}$ corresponding good $B$ values. This requires a work load of about $2^{w-\log_2 w+w+2\log_2 w} = 2^{2w+\log_2 w}$ (e.g. $2^{69}$) operations per $S_0$ guess. We are left with a table of $2^{w+2\log_2 w}$ - e.g. $2^{42}$ - $(D, \Delta A, \Delta C)$ frequencies.

• Next, for every possible $S_1$ value, we can do the same. For a given guess, we select the $2^{w-\log_2 w}$ possible values for $D$ which achieve $f(D+S_1) \equiv 0 \pmod w$ in the first round, and add their frequencies together. We are left with a table of $w^2$ $(\Delta A, \Delta C)$ frequencies, the maximum of which corresponds to the sum of some key bits when the two subkeys are correctly guessed. This step requires an effort of $2^{w+\log_2 w}$ operations for all $(S_0, S_1)$ subkey guesses.

• Once such a table of frequencies of the $(\Delta A, \Delta C)$ values has been obtained, the distinguisher of Section 4 may be applied quite naturally to it. If $(S_0, S_1)$ is the correct subkey guess, one of the frequencies is expected to pass the test, whereas the test is expected to fail when wrong values have been picked. Thus this procedure allows us to recover the first two subkeys using a memory of less than $2 \cdot 2^{2w+2\log_2 w}$ words (e.g. $2^{75}$) and a workload

$$C = 2^w \left(2 \cdot 2^{2w+\log_2 w}\right) = 2^{3w+\log_2 w+1} \quad ,$$

(e.g. $2^{102}$), which is far less than the number of encryptions needed for the distinguisher anyway. However, using this technique, we filter out about $w^2$

plaintext-ciphertext pairs, therefore we have to start off with far more pairs at the beginning of the attack in order to make sure the distinguisher gets enough information after the filtering phase. This leads to a required number of known plaintexts :

$$m \geq 100 w^{4\lfloor \frac{r-1}{2} \rfloor} \log w.$$

Note that for $w = 32$ and $r = 14$, $m$ is about equal to the $2^{128}$ limit.

### 5.2 Improved Approach Without Filtering

As we saw in the last section the fact that we filter pairs for which the first two rotations are zero "costs" a factor $w^2$ in the number of plaintext-ciphertext pairs. We want to avoid this and use only the $n$ pairs required by the distinguisher. We actually guess the first two rotations at the cost of some more memory.

• Let $\beta = f(B) \pmod{w}$ and $\delta = f(D) \pmod{w}$ be the two rotations of the first round. For each of the $w^2$ potential values of $(\beta, \delta)$, we generate a hash table for the frequencies of the tuples

$$\big(B, D, (A_{\text{in}} \ll \delta) \bmod w, A_{\text{out}} \bmod w, (C_{\text{in}} \ll \beta) \bmod w, C_{\text{out}} \bmod w\big)$$

Thus we have $w^2$ tables of size $2^{2w+4\log_2 w}$ (e.g. $2^{84}$) each, giving all the frequencies for the various potential $(\beta, \delta)$ couples of rotations. Note that the generation of such tables may be optimized (avoiding an extra work factor of $w^2$ for each plaintext-ciphertext pair) in a way which will be discussed below.

• Now for every guess of $S_0$, for each of the $w$ possible $\beta$ values, we may select the $2^{w-\log_2 w}$ (e.g. $2^{27}$) $B$ values such that $f(B+S_0) = \beta \bmod w$ and, for each of the $w$ potential $\delta$ values, add together, in the $(\beta, \delta)$ table, the frequencies of those tuples for which the values of $D$, $\Delta A = (A_{\text{out}} - ((A_{\text{in}} \oplus f(B+S_0)) \ll \delta)) \bmod w$, $C_{\text{in}} \ll \beta \bmod w$ and $C_{\text{out}} \bmod w$ are the same. We thus obtain $w^2$ tables providing $(D, \Delta A, C_{\text{in}} \ll \beta \bmod w, C_{\text{out}} \bmod w)$ frequencies, at the expense of a $2^{2w+5\log_2 w}$ (e.g. $2^{89}$) work load per $S_0$ assumption.

• Next, for each guess of $S_1$, for each of the $w$ possible $\delta$ values, we may select the $2^{w-\log_2 w}$ (e.g. $2^{27}$) $D$ values such that $f(D+S_1) = \delta \bmod w$ and, for each of the $w$ potential $\beta$ values, add together, in the $(\beta, \delta)$ table derived at the former step, the frequencies of those $(D, \Delta A, C_{\text{in}} \ll \beta \bmod w, C_{\text{out}} \bmod w)$ tuples for which the values of $\Delta A$ and $\Delta C = (C_{\text{out}} - ((C_{\text{in}} \oplus f(D+S_1)) \ll \beta)) \bmod w$ are the same. By adding up all the $(\Delta A, \Delta C)$ frequencies obtained for all the $(\beta, \delta)$ pairs, we are left with a table of $w^2$ $(\Delta A, \Delta C)$ frequencies which can be used as an input to the distinguisher of Section 4. The distinguisher is expected to succeed only when the two subkeys $S_0$ and $S_1$ are correctly guessed. Thus the above procedure provides the first two subkeys. The work load for this step is about $2^{w+4\log_2 w}$ for each $(S_0,S_1)$ assumption.

**Discussion.** In order to optimize the generation of the $w^2$ $(\beta,\delta)$ tables of the $(B$, $D$, $A_{\text{in}} \lll \delta \bmod w$, $A_{\text{out}} \bmod w$, $C_{\text{in}} \lll \beta \bmod w, C_{\text{out}} \bmod w)$ frequencies, we suggest the following technique. We denote by $A_L$ and $C_L$ (resp $A_H$ and $C_H$) the $w/2+\lfloor\frac{\log_2 w}{2}\rfloor$ (e.g. 18) lowest (resp highest) weight bits of $A_{\text{in}}$ and $C_{\text{in}}$. From the $n$ plaintext-ciphertext pairs used in the attack, we first derive the four tables containing the $(B$, $D$, $A_L$, $A_{\text{out}} \bmod w$, $C_L$, $C_{\text{out}} \bmod w)$, $(B$, $D$, $A_L$, $A_{\text{out}} \bmod w$, $C_H$, $C_{\text{out}} \bmod w)$, $(B$, $D$, $A_H$, $A_{\text{out}} \bmod w$, $C_L$, $C_{\text{out}} \bmod w)$, and $(B$, $D$, $A_H$, $A_{\text{out}} \bmod w$, $C_H$, $C_{\text{out}} \bmod w)$ frequencies. Each of the $w^2$ $(\beta,\delta)$ tables of $(B$, $D$, $A_{\text{in}} \lll \delta \bmod w$, $A_{\text{out}} \bmod w$, $C_{\text{in}} \lll \beta \bmod w$, $C_{\text{out}} \bmod w)$ frequencies can then be deduced from one of the four above tables. This way, we process the $n$ samples only once, and the additional complexity factor of $w^2$ corresponding to all possible choices for $(\beta,\delta)$ will apply essentially to the number of entries in each table, which is about $2^{3w+2\lfloor\frac{\log_2 w}{2}\rfloor+2\log_2 w}$ For example, for $w = 32$ and $n = 2^{118}$, this complexity is about $2^{10}\cdot 2^{110}$ instead of $2^{10}\cdot 2^{118}$.

The complexity of the entire procedure for recovering the first two subkeys $S_0$ and $S_1$ is less than $2\cdot 2^{3w+5\log_2 w}$ (e.g. $2^{122}$).

Once the first two subkeys are found, we can decrypt one round using the data in the previously described tables and may apply the same technique on the next two subkeys. As we go on recovering the extended key piece by piece, the required number of plaintext-ciphertext pairs to make the distinguisher work decreases very fast. Thus the overall complexity of this attack stays well below the effort of an exhaustive search for the key.

## 6 On the Existence of Similar RC5 Statistics

RC6 is an enhancement of the RC5 encryption algorithm. RC5 is characterized by three parameters $w$ (word size ; note that the RC5 block size is $2w$), $r$ (number of rounds ; unlike an RC6 round, an RC5 round consists of two half rounds) and $b$ (number of key bytes).

The following statistical property of RC5 is closely related to the RC6 properties summarised in Section 3 above : if, in $\rho$ consecutive RC5 half rounds, the rotation amounts applied at each second half round are all equal to zero, then after $\rho$ half rounds the $\log_2 w$ lowest weight bits of one of the two plaintext halfes $A$ and

$B$ has been simply added (modulo $w$) with a constant value derived from the key.

The analysis of Section 4 is easy to transpose, to show that $\rho$ half rounds of RC5 can be distinguished from a random permutation using a number $n$ of known plaintexts which stays within a small factor of $w^{2\lfloor \frac{\rho}{2} \rfloor -1}$. For sufficiently low $r$ values, this distinguisher can be used to guess the RC5-$w/r/b$ expanded key, using a number $n$ of known plaintexts which stays within a small factor of $w^{2(r-1)-1}$. However, for usual RC5 parameter choices such as $r = 12$ and a 64-bit block size, the number of available plaintexts is far too low to mount such an attack.

There are some connections beween the above outlined RC5 attack and the RC5 linear attacks mentioned in [5], which require about $4w^{2(r-1)}$ known plaintexts. Both approaches are based related RC5 properties, and the main difference consists in handling $log_2 w$-bit statistics versus binary statistics. We conjecture - but are not fully sure, since we did not check the RC5 key derivation details - that the treatment of $log_2 w$-bit statistics might provide a slight performance improvement over the linear cryptanalysis approach.

## 7 Conclusion

Extending the work presented at the second AES conference, we have shown the existence of a special statistical phenomenon on RC6 which enables to mount a distinguisher attack on up to 13 rounds. As usual, this kind of attack is shown to be convertible into a known plaintext attack which can break up to 14 rounds of RC6 (or equivalently 15 inner rounds with or without post-whitening), requiring about $2^{118}$ known plaintexts, $2^{112}$ memory and a work load of $2^{122}$ operations. Of course this attack is not anywhere near practical, but still leads us to the conclusion that due to the existence of a slight but iterative statistical weakness in its round function, RC6 does not have a very conservative number of rounds.

## References

1. http://www.nist.gov/aes
2. O. Baudron, H. Gilbert, L. Granboulan, H. Handschuh, A. Joux, P. Nguyen, F. Noilhan, D. Pointcheval, T. Pornin, G. Poupard, J. Stern, S. Vaudenay, “Report on the AES Candidates,” *The Second Advanced Encryption Standard Candidate Conference,* N.I.S.T., 1999, pp. 53–67.
3. FIPS 46, *Data Encryption Standard,* US Department of Commerce, National Bureau of Standards, 1977 (revised as FIPS 46–1:1988; FIPS 46–2:1993).
4. T. Iwata, K. Kurosawa, “On the Pseudorandomness of AES Finalists – RC6 and Serpent”, These proceedings.

5. B. S. Kaliski Jr., Y. L. Yin, “On the Security of the RC5 Encryption Algorithm”, RSA Laboratories Technical Report TR-602, Version 1.0 - September 1998.
6. L. Knudsen, W. Meier, “Correlations in RC6 with a reduced number of rounds ”, These proceedings.
7. M. Matsui, “The first experimental cryptanalysis of the Data Encryption Standard”. In *Advances in Cryptology - Crypto’94*, pp 1-11, Springer Verlag, New York, 1994.
8. S. Moriai, S. Vaudenay, “Comparison of randomness provided by several schemes for block ciphers”, Preprint, 1999.
9. R.L. Rivest, M.J.B. Robshaw, R. Sidney and Y.L. Yin,“The RC6 Block Cipher”, v1.1, August 20, 1998.
10. S. Vaudenay, “An experiment on DES - Statistical Cryptanalysis”. In *3rd ACM Conference on Computer Security*, New Dehli, India, pp139-147, ACM Press, 1996.