RC5 Extended
(RC5$\frac{1}{2}$)

Tolga Acar
Utah Valley Cryptologic Society
t.acar@computer.org

Wednesday, March 27, 2002

1 Introduction

This is an attempt to extend the RC5 cipher in order to increase its block size, and to observe its modified behaviour. The extended cipher presented here doubles the block size. The block size is four times the word size w instead of two.

We will refer to the extended RC5 as RC5X. We are not interested in the key expansion and we use the RC5's key expansion to fill the expanded key table S.

2 Parameters and Operations

The wordsize is w in bits, typically 16, 32, or 64. The block size is $4w$ bits ($2w$ in RC5). r is the number of rounds, $0 \leq r \leq 255$. b is the key K length in bytes, and the expanded table S size is $t = 4(r+1)$ in words, it is $2(r+1)$ in RC5.

Operations:

- $+$ is two's complement modulo 2^w addition, carry ignored
- $-$ is the inverse of '+'
- $\oplus$ is bitwise exclusive OR (XOR)
- $\lll$ is left rotation
- $\ggg$ is right rotation

2.1 Key Expansion

We are not interested in modifying the key expansion, we use the RC5's key expansion to fill the expanded key table S. RC5X's expanded key table S is of size $t = 4(r+1)$ words. Like RC5, it uses two magic constants: $P_w = Odd((e-2) * 2^w)$, and $Q_w = Odd((\phi - 1) * 2^w)$.

Below, u is the number of bytes per word, c is the number of words in key K rounded up (ceil). L is a temporary word array of size c. A and B are one word variables. The only difference below is the expanded table size t from the original RC5's key expansion.

```
u = w/8
c = ceil(b/u)
t = 4(r+1)
L[0..c-1] = 0
L[0..c-1] = K[0..b-1]
S[0] = Pw
for i=1 to t-1 do
  S[i] = S[i-1] + Qw
i = j = 0
A = B = 0
do 3*max(t,c) times
  A=S[i]=(S[i]+A+B) << 3
  B=L[j]=(L[j]+A+B) << (A+B)
  i=(i+1) mod t
  j=(j+1) mod c
```

2.2 RC5 Encrypt and Decrypt

For reference and completeness.

Encrypt

Input: A, B each w bits
$A = A + S_0$
$B = B + S_1$
for $i = 1$ to r do
$\quad A = ((A \oplus B) \lll B) + S_{2i+0}$
$\quad B = ((B \oplus A) \lll A) + S_{2i+1}$
Output: A, B each w bits

Decrypt

Input: A, B each w bits
for $i = r$ downto 1 do
$\quad B = ((B - S_{2i+1}) \ggg A) \oplus A$
$\quad A = ((A - S_{2i+0}) \ggg B) \oplus B$
$\quad B = B - S_1$
$\quad A = A - S_0$
Output: A, B each w bits

3 Generalized Structure

In RC5, there are two blocks each one word in a round, and both are processed in the round. Instead of left and right parts of a block in the typical Feistel ciphers, the extended RC5 has more than two parts per block [1]. The round of a Feistel function has the general form of $R_i = L_{i-1} \oplus f(R_{i-1}, S_i)$. Instead of calling the parts left and right (L and R), we call them *parts*, $P^{(k)}$, $k = 0 \ldots u-1$, and u represents the number of parts in each block. Typically, u is a power of 2, but this is not a strict requirement. In the remaining of this study, we assume that u is a power of two, i.e., $u = 2^v$. Then, $v = 1$ represents a Feistel structure, hence $R_i = P_i^{(0)}$ and $L_i = P_i^{(1)}$.

[1] Strictly speaking, RC5 is not a Feistel cipher

3.1 Round Function

In accordance with the strict definition of a Feistel cipher, only one of the u parts is processed in a round. We theorize that increasing the number of parts increases the block size, but requires more rounds to achieve similar security with the two part system.

In order to limit explosion of the number of rounds with extended part count, more than one part can be processed in a round similar to RC5. If all parts are processed in a round, we hypothesize that the number of rounds is one vth ($u = 2^v$) of the number of rounds required in a traditional Feistel structure where one part is processed in a round. For instance, the number of rounds in RC5 is half of its Feistel counterpart.

A u-part RC5 would have a basic part operation

$$P_i^{(k)} = ((\oplus_{j \neq k} P_i^{(j)}) \lll P_i^{(k)}) + S_{u \cdot i + k}$$

where $k = 0 \ldots u$. Note that the result of this computation is assigned to $P_i^{(k)}$ instead of $P_{i+1}^{(k)}$, which means that the order is important. We assume that the lower numbered part is processed before higher numbered parts, e.g., $P_i^{(0)}$ is processed before $P_i^{(1)}$, hence, higher-numbered parts include the modified parts in the same round, presumably increasing the avalance effect.

Extended RC5 with u parts per block can be expressed as

```
for k=0 to u-1 do
   P(0) = IN[k] + S[k]
for i=1 to r do
   for k=0 to u-1 do
      x = 0
      for j=0 to u-1 do
         x = x XOR P(j)
      P(k) = (x <<< P((k+1) mod u) + S[u*i+k]
```

It is important to note that the rotation is based on a part that is not yet updated. This is required for decryption.

```
for i=0 to u-1 do
   P(k) = IN[k]
for i=1 to r do
   for k=u-1 downto 0 do
      x = P(k)
      for j=0 to u-1 do
         x = x XOR P(j)
      P(k) = ((P(k) - S[u*i+k]) >>> P((k-1) mod u)) XOR x
```

Can the part processing order be reversed in both encryption and decryption without loss of generality? (Yes).

4 Four-Part Extended RC5

Our first attempt is to double the RC5's blocksize. That is, $v = 2, u = 4$. Then the encryption and decryption look like the following.

Input: A, B, C, D each w bits
$A = A + S_0$
$B = B + S_1$
$C = C + S_2$
$D = D + S_3$
for $i = 1$ to r do
$A = ((A \oplus B \oplus C \oplus D) \lll B) + S_{4i+0}$
$B = ((B \oplus A \oplus C \oplus D) \lll C) + S_{4i+1}$
$C = ((C \oplus A \oplus B \oplus D) \lll D) + S_{4i+2}$
$D = ((D \oplus A \oplus B \oplus C) \lll A) + S_{4i+3}$
Output: A, B, C, D each w bits

Input: A, B, C, D each w bits
for $i = r$ downto 1 do
$D = ((D - S_{4i+3}) \ggg A) \oplus (A \oplus B \oplus C)$
$C = ((C - S_{4i+2}) \ggg D) \oplus (A \oplus B \oplus D)$
$B = ((B - S_{4i+1}) \ggg C) \oplus (A \oplus C \oplus D)$
$A = ((A - S_{4i+0}) \ggg B) \oplus (B \oplus C \oplus D)$
$D = D - S_3$
$C = C - S_2$
$B = B - S_1$
$A = A - S_0$
Output: A, B, C, D each w bits

5 Timings

The initial C implementation of RC5 and RC5X gave similar throughput at $w = 32$ and $r = 16$. The measured throughput is given in KBytes/msec in the following table.

Cipher	w	Encrypt	Decrypt
RC5	32	31	29
RC5	64	9	8
RC5X	32	29	33
RC5X	64	16	19

6 Analysis - Incomplete

One part encryption operation in round i can be written as

$$
\begin{aligned}
A &= ((A \oplus B \oplus C \oplus D) \lll B) + S_{4i} \\
S_{4i} &= A - ((A \oplus B \oplus C \oplus D) \lll B)
\end{aligned}
$$