The RC5 Encryption Algorithm

A fast, symmetric block cipher that may replace DES

Ronald L. Rivest

The RC5 encryption algorithm is a fast, symmetric block cipher suitable for hardware or software implementations. A novel feature of RC5 is the heavy use of data-dependent rotations. RC5 has a variable-length secret key, providing flexibility in its security level.

RC5 is a parameterized algorithm, and a particular RC5 algorithm is designated as *RC5-w/r/b*. The parameters are as follows:

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

RC5 uses an "expanded key table," S, derived from the user's supplied secret key K. The size t of table S depends on the number r of rounds: S has $t=2(r+1)$ words.

RC5 is not intended to be secure for all possible parameter values. On the other hand, choosing the maximum parameter values would be overkill for most applications.

We provide a variety of parameter settings so that users may select an encryption algorithm whose security and speed are optimized for their application, while providing an evolutionary path for adjusting their parameters as necessary in the future.

For example, RC5-32/16/7 is an RC5 algorithm with the number of rounds and the length of key equivalent to DES. Unlike unparameterized DES, however, an RC5 user can upgrade the choice for a DES replacement to an 80-bit key by moving to RC5-32/16/10.

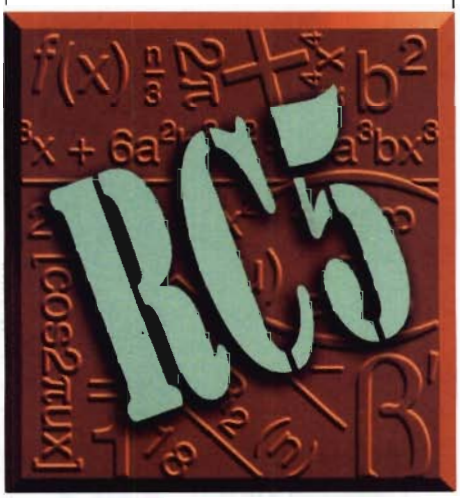
Ron is associate director of the MIT Laboratory for Computer Science, a coinventor of the RSA public-key cryptosystem, and a cofounder of RSA Data Security Inc. He can be contacted at rivest@theory.lcs.mit.edu. RC5 and RSA-RC5 are trademarks of RSA Data Security Inc. Patent pending.

As technology improves, and as the true strength of RC5 algorithms becomes better understood through analysis, the most appropriate parameters can be chosen. We propose RC5-32/12/16 as providing a "nominal" choice of parameters. Further analysis is needed to analyze the security of this choice.

Overview of the Algorithm

RC5 consists of three algorithms, one each for key expansion, encryption, and decryption. These algorithms use the following three primitive operations (and their inverses).

- Two's complement addition of words, denoted by "+". This is modulo-2^w addition.
- Bit-wise exclusive-OR of words, denoted by $\oplus$.
- A left-rotation (or "left-spin") of words: the rotation of word x left by y bits is denoted $x <<< y$. Only the $\lg(w)$ low-order bits of y are used to determine the rotation amount, so that y is interpreted modulo w.



The key-expansion routine expands the user's key K to fill the expanded key array S, so S resembles an array of t random binary words determined by the user's secret key K. The array S is first initialized using a linear congruential generator modulo 2^w determined by some "magic constants." Then, S is mixed with the secret key K in three passes by both the + and <<< operations.

The key-expansion function has a certain amount of "one-wayness": It is not so easy to determine K from S.

For encryption, we assume that the input block is given in two w-bit registers, A and B, and the ouput is also placed in the registers A and B. Example 1 is a pseudocode version of the encryption algorithm. The output is in the registers A and B. The decryption routine is easily derived from the encryption routine.

```
A=A+S[0];
B=B+S[1];
for i=1 to r do
    A=((A⊕B)<<<B)+S[2*i];
    B=((B⊕A)<<<A)+S[2*i+1];
```

Example 1: *Pseudocode of RC5 encryption algorithm.*

Speed and Security

The encryption algorithm is very compact, and can be coded efficiently in assembly language on most processors. The table S is accessed sequentially, minimizing issues of cache size. The RC5 encryption speeds obtainable are yet to be fully determined. For RC5-32/12/16 on a 90-MHz Pentium, a preliminary C++ implementation compiled with the Borland C++ compiler (in 16-bit mode) performs a key setup in 220 µsec and performs an encryption in 22 µsec (equivalent to 360,000 bytes/sec). These timings can presumably be improved by more than an order of magnitude using a 32-bit compiler and/or assembly language — an assembly-language routine for the 486 can perform each round in eight instructions.

A distinguishing feature of RC5 is its heavy use of data-dependent rotations — the amount of rotation performed is dependent on the input data, and is not predetermined.

The use of variable rotations should help defeat differential and linear cryptanalysis since bits are rotated to "random" positions in each round.

I invite the reader to help determine the strength of RC5.

Acknowledgments

I'd like to thank Burt Kaliski, Lisa Yin, Paul Kocher, and everyone else at RSA Laboratories for their comments and constructive criticism.

References

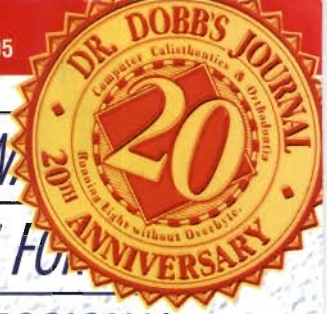
Biham, E. and A. Shamir. *A Differential Cryptanalysis of the Data Encryption Standard.* Berlin: Springer-Verlag, 1993.

Matsui, Mitsuru. "The First Experimental Cryptanalysis of the Data Encryption Standard" in *Proceedings CRYPTO '93.* Berlin: Springer, 1994.

Reprinted with permission from **Dr. Dobbs Journal,** January, 1995, Vol. 20 Issue 1

GOST—Unlocking Soviet Secrets!

#226 JANUARY 1995



Dr. Dobb's JOURNAL

SOFTW... TOOLS FO... PROFESSIONAL PROGRAMMER

NUMERIC PROGRAMMING...

TOOLS THAT COUNT

- PENTIUM OPTIMIZATION
- UNDOCUMENTED FORTRAN
- INFINITE INTEGERS
- RC5 ENCRYPTION
- VIRTUAL REALITY
- INSIDE WINDOWS 95

```
SUBROUTINE ISORT (ARY, N)
integer *2 ary (n), icmpi
external icmpi

call qsort (ary, n, 2, icmpi)

return
end
INTEGER*2 FUNCTION ICMPI
integer*2 x,y
if (x .gt. y) then
  icmpi = +1
else
  icmpi = ø
endif
return
end
SUBROUTINE RSORT (ARY,N)
real*4 ary(n)
integer*2 icmpr
external icmpr

call qsort (ary, n 4, icmpr)

return
end
INTEGER*2 FUNCTION ICMPR
real*4 x,y
if (x .lt. y) then
  icmpr = -1
else if (x .gt. y) then
  icmpr = +1
```

$3.95 ($4.95 CANADA)

01

0 38351 16562 8

A Miller Freeman Publication