The RC5 Encryption Algorithm

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^n addition.
- Bit-wise exclusive-OR of words, denoted by ⊕.
- 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) lower-order bits of y are used to determine the rotation amount, so that y is interpreted modulo w.

Example 1: Pseudo code of RC5 encryption algorithm.

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

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 speed 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 usec and performs an encryption in 22 usec (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 pre-determined.

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

Reprinted with permission from Dr. Dobb’s Journal, January, 1995, Vol. 20 Issue 1 © Copyright 1995 Miller Freeman, Inc. All Rights Reserved.
NUMERIC PROGRAMMING...

TOOLS THAT COUNT

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

$3.95 ($4.95 CANADA)