Performance of Block Ciphers and Hash Functions — One Year Later

Michael Roe

Cambridge University Computer Laboratory,
Pembroke Street, Cambridge CB2 3QG, UK
Email: mrr@cl.cam.ac.uk

1 Introduction

This paper extends the algorithm performance measurements which were presented at the 1993 workshop on fast software encryption [15]. The measurement techniques which were used are described in the original paper [15].

The main changes from the original paper are as follows:

- The NIST Secure Hash Algorithm (SHA) has been replaced with a new algorithm, SHA-1 [10]. The reason for this change is that NIST (or NSA) discovered an attack against the original SHA algorithm [11].
- This year's measurements are based on a faster implementation of GOST 28147.
- This year's measurements were made with a different Sun workstation. The new machine is significantly slower; as a result, all the figures in the "Sparc" column of the tables have changed.
- Some stream ciphers have been included. Many of the most interesting new algorithms in 1994 were stream ciphers. In particular, 1994 saw the publication of what were alleged to be the specifications of two proprietary stream ciphers, RC4 [1] and A5.

2 Apparatus

These measurements were carried out on Unix workstations from two different manufacturers:

- A DEC 3000/400 "Sandpiper". This uses an Alpha CPU clocked at 133 MHz. It has two times 8 KB of primary cache memory and 512 KB of secondary cache.
- A Sparc Station SLC. This uses a Sparc CPU (clock speed unknown).

[1] RC4 is a registered trademark of RSA Data Security Inc.

3 Experimental Approach

The performance test for hash algorithms measures the time taken to hash a message containing 6,400,000 octets. This is implemented by clearing a 64 octet buffer, calling the routine to initialise hashing, calling the routine to hash a buffer 100,000 times, and then calling the routine to finish hashing.

The performance test for symmetric key algorithms measures the time taken to encrypt a message of 6,400,000 bits with a fixed key. This is implemented by clearing a 64-bit buffer, and then calling the routine to encrypt the buffer 100,000 times.

The performance test for stream ciphers measures the time taken to generate 6,400,000 bits of keystream. The different stream cipher implementations generate different amounts of keystream per subroutine call. The algorithms from Bill Chambers generate 32 bits of keystream at a time, while SEAL generates 32768 bits, and WAKE generates 4096 bits. This is shown in the "b/call" column of figure 2. Implementations which only generate a few bits per subroutine call are clearly at a disadvantage in this test; more subroutine calls will be needed to generate the same amount of keystream, and so they will be slower. It should be possible to re-implement Bill Chambers' stream ciphers in such a way that more keystream is generated with each call. This ought to improve their performance.

4 Algorithm Parameters

Some of the algorithms which were tested provide a variable level of security. That is, they have a parameter which the user can set so as to select an appropriate compromise between security and execution speed. For performance measurements of such algorithms to be meaningful, the settings of the parameters must be given. In these experiments, the following parameter settings were used:

RC5	32 bit word size, 32 rounds, 160 bit key
SAFER-K64	6 rounds
Blowfish	64 bit key

Fig. 1. Algorithm Parameters

5 Results

Figure 2 shows the speeds (in Mbits/second) of the algorithms on the two test machines.

Stream Ciphers				
Proposer	Name	b/call	Sparc Mb/s	Alpha Mb/s
David Wheeler	WAKE [20]	4096	14.12	117.0
Phil Rogaway	SEAL [16]	32768	16.64	114.8
Bill Chambers	Clock Controlled [2]	32	2.40	19.2
Ron Rivest	RC4	1024	3.06	15.4
Bill Chambers	Linear Congruential [2]	32	0.031	0.738

Block Ciphers				
Proposer	Name	Block Size	Sparc Mb/s	Alpha Mb/s
Burt Kaliski	[6]	8192	2.87	26.8
Bruce Schneier	Blowfish [18]	64	1.62	11.63
Ron Rivest	RC5 [14]	64	1.42	7.68
Jim Massey	SAFER-K64 [7]	64	1.31	7.68
GOST	GOST 28147 [17, 19]	64	0.75	7.25
Joan Daemen	3WAY [4]	96	0.56	5.24
Meyer,Tuchman	DEA-1 [9]	64	0.294	1.855
Meyer	DEA-1, EDE mode [8]	64	0.168	1.200

Hash Functions				
Proposer	Name	Hash Size	Sparc Mb/s	Alpha Mb/s
Ron Rivest	MD4 [12]	128	9.45	78.77
Ron Rivest	MD5 [13]	128	7.28	60.02
RIPE project	RIPE-MD [3]	128	5.76	48.00
NIST	SHA-1 [10]	160	4.23	41.51
Burt Kaliski	MD2 [5]	128	0.137	0.755

Fig. 2. Algorithm Speeds — Long Messages

References

1. R. J. Anderson, editor. *Fast Software Encryption*, number 809 in Lecture Notes in Computer Science. Springer-Verlag, December 1993.
2. B. Chambers. Two stream ciphers. In Anderson [1], pages 51 – 55.
3. CWI, Amsterdam. *RIPE Integrity Primitives — Final Report of RACE Integrity Primitives Evaluation (R1040)*, June 1992.
4. J. Daemen, R. Govaerts, and J. Vandewalle. A new approach to block cipher design. In Anderson [1], pages 18 – 32.
5. B. Kaliski. *RFC 1319 : The MD2 Message-Digest Algorithm*, April 1992.
6. B. Kaliski and M. Robshaw. Fast block cipher proposal. In Anderson [1], pages 33 – 40.
7. J. L. Massey. SAFER K-64: A byte-oriented block-ciphering algorithm. In Anderson [1], pages 1 – 17.
8. C. H. Meyer and S. M. Matyas. *Cryptography: a new dimension in computer data security.* John Wiley and Sons, 1982.
9. National Bureau of Standards. *Federal Information Processing Standard — Publication 46: Data Encryption Standard*, 1977.
10. National Institute of Standards and Technology. *Federal Information Processing Standard — Publication 180-1: Secure Hash Standard*, May 1994.
11. National Institute of Standards and Technology. NIST announces technical correction to secure hash standard. Press release, April 1994.
12. R. L. Rivest. *RFC 1320 : The MD4 Message-Digest Algorithm*, April 1992.
13. R. L. Rivest. *RFC 1321 : The MD5 Message-Digest Algorithm*, April 1992.
14. R. L. Rivest. The RC5 encryption algorithm. In *Fast Software Encryption*, Lecture Notes in Computer Science. Springer-Verlag, pages 86–96 (these proceedings).
15. M. Roe. Performance of symmetric ciphers and one-way hash functions. In Anderson [1].
16. P. Rogaway and D. Coppersmith. A software optimised encryption algorithm. In Anderson [1], pages 56 – 63.
17. [Russian] State Committee for Standardization, Metrology and Certification. *GOST 28147 : Cryptographic Protection for Data Processing Systems — Cryptographic Transformation Algorithm*, 1990.
18. B. Schneier. Description of a new variable-length key, 64-bit block cipher (Blowfish). In Anderson [1], pages 191 – 204.
19. B. Schneier. The GOST encryption algorithm. *Dr. Dobb's Journal*, pages 143 – 144, January 1995.
20. D. Wheeler. A bulk data encryption algorithm. In Anderson [1], pages 127 – 134.