A Survey of Hardware Implementations of RSA

(Abstract)

Ernest F. Brickell

Sandia National Laboratories *
Albuquerque, NM 87185

Today, a dozen years after the discovery of the RSA encryption algorithm [12], there are many chips available for performing RSA encryption [1] [3] [4] [5] [8] [9] [13] [15]. The purpose of this paper is to briefly describe some of the different computational algorithms that have been used in the chip designs and to provide a list of all of the currently available chips. In this abstract, we will simply mention some of these computational algorithms and give references. The full paper will contain more details of these algorithms and will appear in a book on survey articles in Cryptology which is being edited by Gus Simmons and will be published by IEEE in 1990.

Recall that the RSA encryption function consists of computing $m^e \bmod N$, where $N = pq$ for primes p and q. All of the chips perform the exponentiation as a series of modular multiplications. The modular multiplications are computed either as a standard multiplication followed by a modular reduction, or, more commonly, the computation of the multiplication and the modular reduction is combined. Finally, the multiplications are implemented as a series of additions.

For each of these arithmetic functions, we will mention some choices in how they can be implemented on a chip. By using redundant number systems to avoid carries the addition can be speeded up at a cost of more storage. Multiplication can be speeded up by the techniques of multiple bit scanning. See for instance [6]. There are several techniques that have been developed for implementing modular reduction. The quotient digits can be approximated using only the high order bits of the divisor and the current remainder[3]. Division can be avoided all together by several different methods. The reciprocal of the modulus can be stored, thus replacing division by multiplication. For well chosen values of i, the reduced values of $2^i \bmod N$ can be stored, so that modular reduction can be achieved through multiplication by these values. Peter Montgomery [10] has a method for modular reduction without division which uses a nonstandard technique of identifying the residue classes.

There are also techniques available to save on the number of multiplications needed to perform an exponentiation. Compared with the standard binary method of expo-

*This work performed at Sandia National Laboratories and supported by the U.S. Department of Energy under contract No. DE–AC04–76DP00789.

G. Brassard (Ed.): Advances in Cryptology - CRYPTO '89, LNCS 435, pp. 368-370, 1990.

nentiation, addition chains [7] can give a significant savings in the number of multiplications needed at a cost of increasing the storage necessary. For the user who knows the factorization of N, it is possible to speed up the computation by the use of the Chinese Remainder Theorem[7]. Since some of the chip manufacturers give their speeds assuming the use of the Chinese Remainder Theorem while others do not, it is often difficult to compare the performance of the different chips. The following chart contains all of the actual RSA chips that the author is aware of.

	Year	Tech.	# bits per chip	Clock	Baudrate (# bits)	# Clocks per 512 bit encryption
Sandia	1981	3μm	168	4MHz	1.2K (336)	$4.0 * 10^6$
Bus. Sim.	1985	Gate Array	32	5MHz	3.8K (512)	$.67 * 10^6$
AT&T	1987	1.5μm	298	12MHz	7.7K (1024)	$.4 * 10^6$
Cylink	1987	1.5μm	1024	16MHz	3.4K (1024)	$1.2 * 10^6$
Cryptech	1988	Gate Array	120	14MHz	17K (512)	$.4 * 10^6$
CNET	1988	1μm	1024	25MHz	5.3K (512)	$2.3 * 10^6$
Brit. Telecom	1988	2.5μm	256	10MHz	10.2K (256)	$1 * 10^6$
Plessy	1989		512		10.2K (512)	
Sandia	1989	2μm	272	8MHz	10K (512)	$.4 * 10^6$
Philips	1989	1.2μm	512	16MHz	2K (512)	$4.1 * 10^6$

The last column in this table was estimated if the chips could not do a 512 bit encryption or if the timing for a 512 bit encryption was not available. The 12MHz listed for the AT&T chip is for 1024 bit encryption. For a 512 bit encryption, it runs at 15MHz. AT&T has recently come out with an improved version of their chip which has 520 bit slices per chip and is slightly faster. At first glance, the Philips design does not appear competitive with the others. However, this is a design for smart cards and only takes 4mm^2 of silicon. Cylink and Siemanns are also planning smart card implementations.

There are chip designs that promise much greater speeds than current chips [11, 14], but chips based on these designs have not yet been built.

In recent years, digital signal processors (DSP) have become a viable alternative to building a custom chip for RSA encryption. Kochanski [8] was the first to consider this possibility. DSPs have improved since his work and Michael Weiner of BNR has announced an implementation on the Motorola 56000 that achieves a 125 ms encryption on a 512 bit modulus for a throughput of 4K bits per second without using the Chinese Remainder Theorem. Using the Chinese Remainder Theorem, he can achieve a 50 ms encryption.

Another alternative to custom design has been recently proposed by Bertin, Roncin, and Vuillemin [2]. They implemented RSA on a pair of Programmable Active Memory chips. The maximum modulus size that they can accommodate on two chips is 508 bits. Using the Chinese Remainder Theorem, the encryption time is 17 ms for a baudrate of 30K bits per second.

References

[1] AT&T, *T7002/t7003 bit slice multiplier.* Product Announcement, 1987.

[2] P. BERTIN, D. RONCIN, AND J. VUILLEMIN, *Introduction to programmable active memories.* Internal Report, Digital Equipment Corporation, 1989.

[3] E. F. BRICKELL, *A fast modular multiplication algorithm with applications to two key cryptography*, in Advances in Cryptology, Proceedings of Crypto 82, D. Chaum, R. L. Rivest, and A. T. Sherman, eds., New York, 1982, Plenum Press, pp. 51–60.

[4] P. GALLAY AND E. DEPRET, *A cryptography processor*, in 1988 IEEE International Solid-State Circuits Conference Digest of Technical Papers, 1988, pp. 148–149.

[5] F. HOORNAERT, M. DECROOS, J. VANDEWALLE, AND R. GOVAERTS, *Fast RSA-hardware: Dream or reality?*, in Advances in Cryptology-EUROCRYPT'88, C. G. Günther, ed., 1988.

[6] K. HWANG, *Computer Arithmetic*, John Wiley, New York, 1979.

[7] D. KNUTH, *The art of computer programming, Vol. 2 : Seminumerical algorithms*, Addison-Wesley, Reading,MA, 1981.

[8] M. KOCHANSKI, *Developing an RSA chip*, in Advances in Cryptology-CRYPTO'85, H. C. Williams, ed., New York, 1985, Springer-Verlag, pp. 350–357.

[9] S. MIYAGUCHI, *Fast encryption algorithm for the RSA cryptographic system*, in Proceedings of Compcon 82, Los Angeles, 1982, IEEE, pp. 672–678.

[10] P. L. MONTGOMERY, *Modular multiplication without trial division*, Mathematics of Computation, 44 (1985), pp. 519–521.

[11] G. ORTON, M. ROY, P. SCOTT, L. PEPPARD, AND S. TAVARES, *VLSI implementation of public-key encryption algorithms*, in Advances in Cryptology-CRYPTO'86, A.M.Odlyzko, ed., New York, 1986, Springer-Verlag, pp. 277–301.

[12] R. RIVEST, A. SHAMIR, AND L. ADLEMAN, *A method for obtaining digital signatures and public-key cryptosystems*, Communications of the ACM, 21 (1978), pp. 120–126.

[13] R. L. RIVEST, *RSA chips (past/present/future)*, in Advances in Cryptology-EUROCRYPT'84, T.Beth, N.Cot, and I.Ingemarsson, eds., New York, 1984, Springer-Verlag, pp. 159–168.

[14] H. SEDLAK AND U. GOLZE, *An RSA cryptography processor*, Microprocessing and Microprogramming, 18 (1986), pp. 583–590.

[15] A. VANDEMEULEBROECKE, E. VANZIELEGHEM, T. DENAYER, AND P. G. JESPERS, *A single chip 1024 bits RSA processor.* to appear in Advances in Cryptology - EUROCRYPT'89.