CWI Quarterly Volume 4 (1) 1991, pp. 69 – 72

Factorization beyond the Googol with MPQS on a Single Computer

Herman te Riele, Walter Lioen & Dik Winter
CWI
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

For the first time a number of more than 100 decimal digits has been factorized on a single computer by means of the Multiple Polynomial Quadratic Sieve method of Kraïtchik and Pomerance (with improvements by Montgomery and Silverman). This method (MPQS) is the best one known to handle numbers which are the product of two large, approximately equal prime factors. These numbers are being used in cryptography as keys in public-key cryptosystems. The safety of such cryptosystems depends on our ability to factorize these keys.

The computer used is the four-processor Cray Y-MP4/464 which was installed at SARA (The Academic Computing Centre Amsterdam) in December 1990. The job was started on January 18, 1991, was finished on February 11, 1991 (the day that CWI celebrated its 45-th birthday) and consumed about 475 CPU-hours. It ran on one processor of the four-processor Cray Y-MP4. If four processors would have been used, the job could have been completed in about five days. The previous single-computer record with MPQS was a 92-digit number, factorized in the spring of 1988 on a NEC SX-2 computer ([6]).

The version of MPQS we used is described in [8] and [6]. It was not very difficult to optimize our portable Fortran program on the Cray Y-MP4. The sieve loop (cf. [6, p. 46]) was vectorized automatically, and we measured an average speed of about 110 million additions per second (against 90 million on the NEC SX-2). In the selection loop (cf. [6, pp. 47–48]) we reached a speed of about 150 million comparisons per second, by using the Cray SCILIB-routine `WHENFGT` (against 90 million on the NEC SX-2).

We were allowed to run our program for free before February 20, 1991, the day the Cray Y-MP4 was officially put into use. Therefore, we did not yet find the time to incorporate in our program the recent improvements proposed by Lenstra and Manasse [5] (who not only allow *one*, but also *two* large primes in the incomplete relations) and by LaMacchia and Odlyzko [3] (who propose a special technique called *Structured Gaussian elimination* to reduce the size of the large sparse matrix involved in the MPQS algorithm; this technique turned

out to be very efficient in the factorization of the ninth Fermat number [4]). We hope to implement and test these improvements in the near future: we estimate that this will at least double the speed of our program (for 100-digit numbers).

The number we have factorized belongs to the present list of 'more wanted' Cunningham numbers [1]. It consists of 101 decimal digits, and can be written as

$$\frac{2^{463}+1}{3 \times 2356759188941953 \times 76834966209858049526107}.$$

The three prime factors given here were known already. With the Cray Y-MP4 we found the two remaining prime factors (of 35 and 66 digits, resp.)

$$88119307925269041107418404833666787$$

and

$$49753260455140380065971880516533368559591310618679245402699521013 9.$$

A short test run on the SX-2 showed that on this computer we could factorize this number in about 550 hours.

In 1984, Davis and Holdridge [2] announced the factorization of the 71-digit composite number R71 $= (10^{71}-1)/9$ in 9.5 hours on a Cray X-MP by means of the first (as far as we know) working version of the quadratic sieve algorithm (called the special q's variation). Based on this result, Pomerance [7] predicted that the Sandia group could factor any 101-digit number in a year-long run on the Cray X-MP. We estimate that the corresponding time on the Cray Y-MP4 would be about nine months. The reduction we found from nine months to 20 days can be explained by the algorithmic improvements found in the meantime ([7], [9]) and the larger central memory available on the Cray Y-MP4/464.

Before we started to factorize the 101-digit number, we tested our program on various smaller numbers. First we took R71 with a factor base consisting of 24,510 elements: the time needed was 0.55 CPU-hours. Next we took a set of 45 numbers of 71 and 72 digits as a contribution to a project of R.P. Brent which aims to extend the Cunningham table. The time needed varied between 0.5 and 1.0 CPU-hours per number. For each number we used a factor base size of about 24,500 elements. Finally, we tried a 81- and a 82-digit number, sent to us by Bob Silverman. Each of them was factorized in about 10 CPU-hours, where we used a factor base size of about 24,500 for both numbers, and where we used the multipliers 5 and 11, respectively. The 81-digit number can be written as

$$\frac{2^{191}+3^{191}}{5 \times 16427 \times 705937}.$$

We found the prime factors (of 29 and 52 digits, resp.)

$$41400945999632457000594230017$$

and

$$562153838216646854022558998721554688424511512759577 3.$$

The 82-digit number can be written as

$$\frac{21^{91}+1}{2\times 11\times 81867661\times 7021471715414521}.$$

For this we found the prime factors (of 40 and 42 digits, resp.)

$$1737708201406774525206464521323498021877$$

and

$$9811835551758002889573950677362390798925 43.$$

In order to facilitate a comparison of our results on the NEC SX-2 with those on the Cray Y-MP4 we give here some details about the two computers, the values of the algorithmic parameters and some results.

Comparison of results with MPQS
on the NEC SX-2 and on the Cray Y-MP4

computer	NEC SX-2	Cray Y-MP4/464
central memory	128 Mbytes	64 Mwords
wordlength	32 bits	64 bits
factorized number is cofactor of	$6^{131}-1$	$2^{463}+1$
# decimal digits	92	101
multiplier used	1	1
factor base bound	600,000	1,300,000
length of sieving interval	5,000,000	9,000,000
bound on the large primes allowed in incomplete relations	30,000,000	65,000,000
smallest sieve prime	31	41
# primes in the factor base	24,334	50,179
# complete relations	9,456	20,428
# incomplete relations	126,094	254,802
# complete relations derived from incomplete relations	14,879	29,755
total CPU-time in hours	95	475 (on one processor)
Gaussian elimination time in hours	0.1	0.5

Acknowledgements

We like to thank the Foundation National Computing Facilities (NCF) for the provision of the computing time on the Cray Y-MP4, and SARA and Cray Nederland BV for the operational support.

References

1. J. Brillhart, D.H. Lehmer, J.L. Selfridge, B. Tuckerman, and S.S. Wagstaff, Jr. *Factorizations of $b^n \pm 1, b = 2, 3, 5, 6, 7, 10, 11, 12$ up to high powers*, volume 22 of *Contemporary Mathematics Series*. American Mathematical Society, Providence, second edition, 1988.
2. J.A. Davis, D.B. Holdridge, and G.J. Simmons. Status report on factoring (at the Sandia National Laboratories). In T. Beth, N. Cot, and I. Ingemarsson, editors, *Advances in Cryptology (Proceedings of EUROCRYPT 84)*, volume 209 of *Lecture Notes in Computer Science*, pages 183–215, Berlin, 1985. Springer.
3. B.A. LaMacchia and A.M. Odlyzko. Solving large sparse linear systems over finite fields. In A.J. Menezes and S.A. Vanstone, editors, *Advances in Cryptology – CRYPTO '90 Proceedings*, volume 537 of *Lecture Notes in Computer Science*, pages 109–133, Berlin, 1991. Springer.
4. A.K. Lenstra, H.W. Lenstra, Jr., M.S. Manasse, and J.M. Pollard. The factorization of the ninth Fermat number. *Mathematics of Computation*, 61(203):319–349, July 1993.
5. A.K. Lenstra and M.S. Manasse. Factoring with two large primes. In I.B. Damgård, editor, *Advances in Cryptology – EUROCRYPT '90 Proceedings*, volume 473 of *Lecture Notes in Computer Science*, pages 72–82, Berlin, 1991. Springer.
6. W.M. Lioen, H.J.J. te Riele, and D.T. Winter. Optimization of the MPQS-factoring algorithm on the Cyber 205 and the NEC SX-2. *Supercomputer 26*, V(4):42–50, July 1988.
7. C. Pomerance, J.W. Smith, and R. Tuler. A pipeline architecture for factoring large integers with the quadratic sieve algorithm. *SIAM Journal on Computing*, 17(2):387–403, April 1988.
8. H.J.J. te Riele, W.M. Lioen, and D.T. Winter. Factoring with the quadratic sieve on large vector computers. *Journal of Computational and Applied Mathematics*, 27(1&2):267–278, September 1989.
9. R.D. Silverman. The multiple polynomial quadratic sieve. *Mathematics of Computation*, 48(177):329–339, January 1987.