Factorization of a 1061-bit number by the Special Number Field Sieve

Greg Childers
California State University Fullerton
Fullerton, CA 92831

August 4, 2012

Abstract

I provide the details of the factorization of the Mersenne number $2^{1061} - 1$ by the Special Number Field Sieve. Although this factorization is easier than the completed factorization of RSA-768, it represents a new milestone for factorization using publicly available software.

1 Introduction

The Number Field Sieve (NFS) is currently the fastest classical algorithm for factoring a large integer into its prime cofactors [8]. Continued study of the practical implementations of the NFS is of significant interest for the security assessment of common public-key cryptosystems, chief among them being the RSA algorithm. The security of the RSA encryption algorithm relies on the fact that integer factorization is difficult. Improvements to the NFS algorithm are of significant practical importance, and factoring milestones are followed by the applied cryptography community. Significant milestones include the factoring of a 512-bit RSA modulus by the general NFS (GNFS) in 2000 [4], $2^{1039} - 1$ by the special NFS (SNFS) in 2007 [2], and a 768-bit RSA modulus in 2010 [7].

Using the SNFS, the complete factorization of the Mersenne number, $2^{1061} - 1$, has been determined. Prior this this effort, this number had no known factors. Although easier than the factorization of RSA-768, this represents a new largest factorization using SNFS, and the largest factorization to date using publicly available software. NFS is comprised of five basic steps: polynomial selection, sieving for relations, filtering of relations, linear algebra, and square root. Each of these steps will be detailed below.

2 Polynomial selection

Polynomial selection consists of finding two polynomials that share a common root modulo the number being factored. For this SNFS factorization, selection of appropriate polynomials is trivial. The polynomials $f(x) = x^6 - 2$ and $g(x) = x - 2^{177}$ share the common root 2^{177} modulo $2^{1061} - 1$.

3 Sieving

Sieving was by far the most computationally intensive step in this factorization. Although sieving is somewhat memory intensive, requiring one to two gigabytes per process, the individual sieving tasks do not need to interact. Taking advantage of the increasing amount of memory in today's typical home computer, I administer a distributed computing project to provide the capacity computing required for the sieving. Using the Berkeley Open Infrastructure for Network Computing (BOINC) [1], the NFS@Home project distributes the NFS sieving tasks to volunteer computers and collects the results [5].

The sieving binaries used are based on the gnfs-lasieve lattice sieve code written by Jens Franke and Thorsten Kleinjung [6] adapted for use in the BOINC framework. Factor base bounds of 250 million were used on both the algebraic and rational sides to reduce the amount of memory required for the computation. A large prime bound of 2^{33} was used on both sides with up to two large primes allowed on the algebraic side and up to three large primes on the rational side. Sieving was performed over a rectangular region of size $2^{16} \times 2^{15}$ using the gnfs-lasieve4I16e binary. These sieving parameters turned out to be suboptimal, requiring sieving over a significantly larger range of q than initially expected. Most of the special-q between 20 million and 1.45 billion on the rational side, and between 20 million and 1.07 billion on the algebraic side were sieved. The total computational effort expended on sieving was approximately 3 CPU-centuries and yielded $671,385,523$ unique relations.

4 Filtering

Filtering was performed using the MSIEVE software library [10]. Filtering started with the set of approximately 671 million unique relations. Following the singleton and clique removal steps, the matrix had approximate 282 million rows. Following the merge phase, a final matrix of 90.3 million rows and columns with an average weight of 125.27 per column was produced. Filtering required approximately 50 CPU-hours and 40 gigabytes of memory to complete, and the final matrix in binary form requires approximately 40 gigabytes of

storage.

5 Linear algebra

The MSIEVE software library uses the block Lanczos algorithm [9] for the linear algebra. In collaboration with Jason Papadopoulos, the author of the MSIEVE library, and Ilya Popovyan [11], a parallel version of MSIEVE utilizing MPI has been written and continues to be optimized. The linear algebra was performed using a 24×24 MPI grid (576 total cores), and was split between the Trestles cluster at the San Diego Supercomputing Center and the Lonestar cluster at the Texas Advanced Computing Center due to allocation details. On both clusters the nodes use DDR3 memory, and communication between nodes within the cluster uses Infiniband interconnects. The linear algebra required approximately 35 CPU-years, and yielded 32 non-trivial dependencies.

6 Square root

MSIEVE uses a straightforward, but memory-intensive, algorithm for the algebraic square root. This involves multiplying all the relations involved in a non-trivial dependency from the linear algebra modulo the monic algebraic polynomial $f(x)$. Because of the size of the products, FFT-based multiplication is used. For $2^{1061} - 1$, this product completed in 9 CPU-hours, required 14 gigabytes of memory, and produced a fifth degree polynomial with coefficients 4.03 gigabits in size. The algebraic square root, calculated using q-adic Newton iteration, required 13.5 CPU-hours. Each dependency has a 50% chance of yielding the factors, and the factors were found on the second dependency. The square root step required a total of 45 CPU-hours.

7 Results

The number $2^{1061} - 1$ is the product of 143-digit and 177-digit prime numbers, $P143 \cdot P177$, where

$$P143 = 46817226351072265620777670675006972301618979214252832875068976303839400413682313921168154465151768472420980044715745858522803980473207943564433$$

$$P177 = 527739642811233917558838216073534609312522896254707972010583175760467054$$
$$896492872702786549764052643493511382273226052631979775533936351462037464$$
$$331880467187717179256707148303247$$

The factorizations of $P143 \pm 1$ and $P177 \pm 1$ are

$$P143 - 1 = 2^4 \cdot 3 \cdot 13 \cdot 29 \cdot 1061 \cdot 1717297 \cdot 2130134834354231 \cdot 5879064877797191 \cdot$$
$$113382740713189708770977555625503236436481111748875743652757880942902$$
$$36117206177559519566271037163 1$$
$$P143 + 1 = 2 \cdot 71 \cdot 229 \cdot 307 \cdot 180073 \cdot 1695653 \cdot 802019957507 \cdot$$
$$60854868590560169873679751115717379719685541 \cdot$$
$$31468597789158489386360641100276762502815436597091702138816877661740 3$$
$$P177 - 1 = 2 \cdot 3^2 \cdot 1061 \cdot 749005237 \cdot 495208476776963 \cdot 662644764100955663 \cdot$$
$$2141325263036607091 \cdot 77577336863559161196624053050198 1 \cdot$$
$$67680043184905880552204706074103597590035826155716962268183844274369428448480629$$
$$P177 + 1 = 2^4 \cdot 267493 \cdot 140551319 \cdot 4927926989 \cdot 342953555423061833 \cdot$$
$$51910213451521968976210662286841720932876616060901913785569590786133054780118355270282012713002975093692818672876614920099431975550440 7$$

These factors will be published in the fourth edition of the Cunningham book [3] and are currently available in the factorization tables of the Cunningham project website [12].

8 Acknowledgments

This calculation could not have been performed without the free, publicly available lattice sieve of Franke and Kleinjung, MSIEVE by Jason Papadopoulos, and the BOINC infrastructure. Computational time on the Trestles and Lonestar clusters is provided by National Science Foundation XSEDE grant number TG-DMS100027. And finally, this effort would not have been possible without the over 1,000 individuals who donate time on their personal computers to the NFS@Home project.

References

[1] D. P. Anderson. BOINC: a system for public-resource computing and storage. In *5th IEEE/ACM International Workshop on Grid Computing*, pages 4–10, 2004.

[2] K. Aoki, J. Franke, T. Kleinjung, A. K. Lenstra, and D. A. Osvik. A kilobit special number field sieve factorization. In *Advances in Cryptology – ASIACRYPT 2007*. Springer, Berlin / Heidelberg, 2007.

[3] 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*. American Mathematical Society, Providence, Rhode Island, third edition, 2002. Available online at http://www.ams.org/publications/online-books/conm22-index.

[4] S. Cavallar, B. Dodson, A. K. Lenstra, W. M. Lioen, P. L. Montgomery, B. Murphy, H. J. J. te Riele, K. Aardal, J. Gilchrist, G. Guillerm, P. C. Leyland, J. Marchand, F. Morain, A. Muffett, C. Putnam, C. Putnam, and P. Zimmermann. Factorization of a 512-bit RSA modulus. In *Advances in Cryptology – EUROCRYPT 2000*. Springer, Berlin / Heidelberg, 2000.

[5] G. Childers. NFS@Home, 2012. http://escatter11.fullerton.edu/nfs/.

[6] J. Franke and T. Kleinjung. Continued fractions and lattice sieving. Proceedings of SHARCS 2005, 2005. Source available for download from http://mersenneforum.org/showthread.php?p=169889.

[7] T. Kleinjung, K. Aoki, J. Franke, A. Lenstra, E. Thomé, J. Bos, P. Gaudry, A. Kruppa, P. Montgomery, D. A. Osvik, H. te Riele, A. Timofeev, and P. Zimmermann. Factorization of a 768-bit RSA modulus. Cryptology ePrint Archive, Report 2010/006, 2010.

[8] A. K. Lenstra and H. W. Lenstra, Jr. The development of the number field sieve. In *Lecture Notes in Mathematics, Volume 1554*. Springer-Verlag, Berlin / Heidelberg, 1993.

[9] P. L. Montgomery. A block Lanczos algorithm for finding dependencies over GF(2). In *EUROCRYPT'95: Proceedings of the 14th annual international conference on theory and application of cryptographic techniques*, pages 106–120, Berlin / Heidelberg, 1995. Springer-Verlag.

[10] J. Papadopoulos. MSIEVE, 2012. http://sourceforge.net/projects/msieve/.

[11] I. Popovyan. Efficient parallelization of Lanczos type algorithms. Cryptology ePrint Archive, Report 2011/416, 2011. http://eprint.iacr.org/.

[12] S. S. Wagstaff, Jr. The Cunningham Project, 2012. http://homes.cerias.purdue.edu/~ssw/cun/.