Factorisation of RSA-220 with CADO-NFS

Shi Bai, Pierrick Gaudry, Alexander Kruppa, Emmanuel Thomé, Paul Zimmermann

FACTORISATION OF RSA-220 WITH CADO-NFS

SHI BAI, PIERRICK GAUDRY, ALEXANDER KRUPPA, EMMANUEL THOMÉ AND PAUL ZIMMERMANN

Abstract. We give details of the factorization of RSA-220 with CADO-NFS. This is a new record computation with this open-source software.

We report on the factorization of RSA-220 (220 decimal digits), which is the 3rd largest integer factorization with the General Number Field Sieve (GNFS), after the factorization of RSA-768 (232 digits) in December 2009 [3], and that of $3^{697}+1$ (221 digits) in February 2015 by NFS@home.

1. Polynomial Selection

Polynomial selection started in December 2013. We used the following degree-6 polynomial:

$$\begin{aligned} f(x) &= 102261713400x^6 \\ &+ 27234181594909908x^5 \\ &- 9930804435111706484585 0x^4 \\ &- 215620030750946901664636580 97x^3 \\ &+ 11805000184948451601984776362928407x^2 \\ &+ 630615566214319854581092023097648620621x \\ &- 57316872946990233765204879797094502162915 25 \end{aligned}$$

with 6 real roots, skewness 330880, Murphy's $\alpha = -9.12$, and Murphy E-value $7.96 \cdot 10^{-13}$ — with parameters $B_f = 5 \cdot 10^8$, $B_g = 2.5 \cdot 10^8$, and area $1.1 \cdot 10^{18}$ — together with the linear polynomial

$$g(x) = 5030366557070505269159x - 52974922542293279416678326382248522.$$

This polynomial pair was found in March 2014.

When the sieving was almost done, we found a slightly better polynomial pair with the new algorithm from [1]:

$$\begin{aligned} f(x) &= 85241880x^6 \\ &- 793823840976030x^5 \\ &- 6158242858878683215870x^4 \\ &+ 43258630815349625184779003 99x^3 \\ &+ 62580417337216766076810348853218424x^2 \\ &+ 39075432077496839033383105327015980460420x \\ &- 93992354401369774621859099268670517220242998848 \\ g(x) &= 3102467199990632324033x - 20738016977319328134879949473884 7537, \end{aligned}$$

with 4 real roots, skewness 3129987, Murphy's $\alpha = -9.17$, and Murphy E-value $8.27 \cdot 10^{-13}$ with same parameters as above.

Date: May 10th, 2016.

2. Sieving

Sieving started in May 2014. We used as factor base bound $500,000,000$ on the rational side, $800,000,000$ on the algebraic side, 2^{34} as large prime bound on both sides, with up to two large primes on the rational side and three on the algebraic side. We used a sieving region of 2^{31} ($I = 16$) like for the factorization of RSA-768.

We used lattice sieving, with special-q's on the algebraic side, and special-q range of 800M-3200M. The average number of relations per special-q dropped from 18.2 at 800M to 11.7 at 3200M, while the average time per relation increased from 5.87s to 9.22s.

In September 2014, we had collected 1,559,374,529 raw relations. We estimate to 370 CPU years the total sieving time (most of the computers used for sieving were Intel Xeons E5-2650 running at 2Ghz).

3. Filtering

The 1,559,374,529 raw relations gave 1,170,018,015 unique relations (i.e., 25% of duplicates). After removing singletons, we had a matrix with about 613M rows, and an initial excess of about 19M. After the "clique removal" strategy from [2], it remained about 496M rows and columns with an excess of 160, and an average weight of 18.48 per row.

In the merge phase, we merged ideals of weight up to 25. The final matrix had about 127M rows and columns, with an average of 175 non-zero coefficients per row. This average density was selected among several (125, 150, 175, 200), based on which gave the best projected linear algebra timings.

4. Linear Algebra

The linear algebra was started on October 8, 2014. The block Wiedemann algorithm has been used, using four sequences of 64-bit vectors (more precisely, blocking parameters $m = 512$ and $n = 256$ have been used for the algorithm).

The first phase (`krylov`) ended on October 20. Four parallel runs of 8-node jobs were used (one for each sequence), each node running on 16 cores. Inter-node communication is achieved with OpenMPI, using Infiniband FDR 56 Gbps interconnect. The core operation of the `krylov` phase, which is the product of the sparse matrix by a vector of 64-bit entries, required 1.39 seconds of wall-clock time on 8 nodes, and was run repeatedly 786,432 times for each sequence.

The second phase (`lingen`) was run in January 2016, and took 42 hours, parallelized on 56 cores (on average, the CPU utilization was 2600%). The memory required was 400 GB. The delay in running the `lingen` is due to the code improvements which were a necessary condition for completing the calculation.

The last phase (`mksol`) was run from April 26 to May 6, 2016, producing 256 kernel vectors (spanning a kernel subspace of dimension 192).

5. Characters and Square Root

The `characters` program was run on May 9, 2016, and gave 146 dependencies. We ran 16 dependencies on May 10, on 16 nodes with 64GB of RAM. Nine of the dependencies gave non-trivial factors.

6. Conclusion

The factorization of RSA-220 is $n = p \cdot q$ where

$$\begin{aligned} p &= 6863656412267566274382371499288437800130842239979164844621244993321541061441464266793821364420842019205499968 7 \\ q &= 3292907439486349812049301549212935291916455196536233952462686051169290349309465246333782486639073819176571260 3 \end{aligned}$$

have both 110 digits.

The prime factors of $p \pm 1$ and $q \pm 1$ are:

$$\begin{aligned} p-1 &= 2 \cdot 13 \cdot 43 \cdot 28193842369532636782383767843087334604038997195313 \cdot p58 \\ p+1 &= 2^3 \cdot 3^2 \cdot 7 \cdot 1249 \cdot 554875542120030541865991790858142414985052193 \cdot p60 \\ q-1 &= 2 \cdot 169219 \cdot 543519485463084901 \cdot 52057548312320557 \cdot 10794188103674435582857519 \cdot p45 \\ q+1 &= 2^2 \cdot 3^2 \cdot 277 \cdot 187027 \cdot 1975255839936851567 \cdot 12171662654885116557914663944896695814 7 \cdot p44 \end{aligned}$$

Acknowledgements. The computation was done with CADO-NFS [4], using mostly a cluster funded by ANR CATREL (`http://catrel.loria.fr/`), and a server machine funded by Région Lorraine, French Ministry of Research, INRIA, CNRS, and the European fund FEDER.

References

[1] Bai, S., Bouvier, C., Kruppa, A., and Zimmermann, P. Better polynomials for GNFS. *Mathematics of Computation 85* (2016), 861–873.

[2] Bouvier, C. The filtering step of discrete logarithm and integer factorization algorithms. `http://hal.inria.fr/hal-00734654`, 2013. Preprint, 22 pages.

[3] Kleinjung, T., Aoki, K., Franke, J., Lenstra, A. K., Thomé, E., Bos, J. W., Gaudry, P., Kruppa, A., Montgomery, P. L., Osvik, D. A., te Riele, H., Timofeev, A., and Zimmermann, P. Factorization of a 768-bit rsa modulus. In *CRYPTO 2010 Advances in Cryptology - CRYPTO 2010* (Santa Barbara, USA, 2010), T. Rabin, Ed., vol. 6223 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 333–350.

[4] The CADO-NFS Development Team. CADO-NFS, an implementation of the number field sieve algorithm. `http://cado-nfs.gforge.inria.fr/`, 2015. Release 2.2.0.