Final Project: Factoring and Primality Testing

Written by Jonathan Hanke

February 14, 2005

Abstract

The problem of finding large prime numbers has always been an interesting pastime, but with the recent advent of public-key cryptography and its implications for data-security and privacy, it has become a very serious endeavor. In the last 30 years, many algorithms have been developed to help answer the questions: ”Is a given number m prime?” and if not, ”What is its prime factorization?”. Answering these questions efficiently involves many of the structures we have considered in class (Z_p, continued fractions, quadratic number fields), as well as some ideas which are a bit deeper (Elliptic curves, complex multiplication, general number fields), and it is interesting to see how much better they allow us do. The prime numbers one can generate at present routinely have hundreds of decimal digits, and in some special cases they can be quite a bit larger. (The largest known prime as of today has 7,235,733 decimal digits!) Your project should explain the answers to these questions, explaining both why these varied methods work, and how efficient they are. It will also be fun to apply them to some large examples to see how practical they are! **[3.5-5]**

Good general references for the arithmetic of elliptic curves are [5], [7]. The topics of factoring and primality testing are the focus of the 2 books [3] and [9], though [9] this is not currently in the Duke library. These are also discussed in [4, Chapter VIII], [5, pp107-118 for ranks for some CM curves, pp148–150 for $\mathbb{Z}_p$ points for some CM curves], [7, pp89–98 for some explicit examples, and especially pp125–138 of Chapter IV for an elliptic curve factoring algorithm]. See also [2, Part III]. An easy computer language to do arbitrary precision integer arithmetic is PYTHON, and some simpl examples are described in [1]. There is also another slightly less friendly program [8] designed specifically for number theory/Elliptic curve computation. Both programs are freely available. There are lots of general crypto refrerences online. One ”standard” refrence is [6], but there are lots of others. The PGP documentation is particularly friendly.

The following is a rough outline which may be useful in thinking about/organizing your project. If you have any questions about your project and/or readings, feel free to let me know, and we can setup a time to talk about it. Have Fun! =)

1. **The brute force approach to factoring**
2. **Fermat’s primality test using $\mathbb{Z}_p$**
3. **Factoring via Pollard ρ and Pollard $p - 1$ methods**
4. **The Quadratic Sieve**
5. **Convergents and Continued fractions**
6. **The Continued fractions algorithm**
7. **Basics about Elliptic Curves**
8. **Elliptic Curve Factorization**
9. **Finding Primitive Roots**
10. **Large Mersenne primes and even perfect numbers**

References

[1] Computations in number theory using python: A brief introduction.

[2] A. K. Bhandari, D. S. Nagaraj, and B. Ramakrishnan, editors. *Elliptic curves, modular forms and cryptography*, New Delhi, 2003. Hindustan Book Agency.

[3] David M. Bressoud. *Factorization and primality testing.* Undergraduate Texts in Mathematics. Springer-Verlag, New York, 1989.

[4] H. Davenport. *The higher arithmetic.* Cambridge University Press, Cambridge, seventh edition, 1999. An introduction to the theory of numbers, Chapter VIII by J. H. Davenport.

[5] Anthony W. Knapp. *Elliptic curves*, volume 40 of *Mathematical Notes.* Princeton University Press, Princeton, NJ, 1992.

[6] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of applied cryptography.* CRC Press Series on Discrete Mathematics and its Applications. CRC Press, Boca Raton, FL, 1997. With a foreword by Ronald L. Rivest.

[7] Joseph H. Silverman and John Tate. *Rational points on elliptic curves.* Undergraduate Texts in Mathematics. Springer-Verlag, New York, 1992.

[8] The PARI Group, Bordeaux. *PARI/GP, version* `2.1.5`, 2004. available from `http://pari.math.u-bordeaux.fr/`.

[9] Song Y. Yan. *Primality testing and integer factorization in public-key cryptography*, volume 11 of *Advances in Information Security.* Kluwer Academic Publishers, Boston, MA, 2004.