Final Project: Elliptic Curve Cryptography

Written by Jonathan Hanke

February 14, 2005

Abstract

The modern phenomenon of public-key cryptography and the role of number theory in designing (hypothetically) secure cryptosystems has given new importance to classical questions like "How can we tell is a number is prime?" and "How can we factor a given number quickly?". The efficiency of the answers determine the "security" of certain modern cryptosystems, which are widely used today. The bad news is that since we don't actually know the best possible answers to these questions, we can't actually guarantee that these commonly used systems are secure. This has driven the search for a variety of different cryptosystems, each of which may have different vulnerabilities and strengths. One recent favorite is to use the points on certain "Elliptic curves" $E : y^2 = x^3 + Ax + B$ for given $A, B \in \mathbb{Z}$ to encrypt information and factor numbers. Your project should describe the basic properties of these curves, explain how they can be useful for encryption and factoring, and do a few examples to illustrate this. **[4-5]**

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. **Diffie-Hellman, El Gamal, and the structure of $\mathbb{Z}_p$**
2. **The arithmetic of rational points on elliptic curves E**
3. **Diffie-Hellman and El Gamal on E over $\mathbb{Z}_p$**
4. **Elliptic curves with complex multiplication**
5. **Elliptic curve factorization and primality testing**

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.