SCHOOF'S ORIGINAL ALGORITHM IS PRACTICAL FOR ELLIPTIC CURVES OF CRYPTOGRAPHIC SIZE

NIKKI PITCHER

In 1985, Schoof's algorithm for counting points on elliptic curves was introduced. It is widely believed that Schoof's original algorithm is not practical for use with elliptic curves of cryptographic size — currently between 160 bits to 256 bits. For example, consider the following statements:

> Schoof, "Counting points on elliptic curves over finite fields," Journal de Théorie des Nombres de Bordeaux **7** (1995), page 219: "This deterministic polynomial time algorithm was impractical in its original form."
>
> Couveignes, Dewaghe, and Morain, "Isogeny cycles and the Schoof-Elkies-Atkin algorithm," LIX/RR/96/03 (1996), page 1: "From a practical point of view, the problem is the size of the torsion polynomials. Indeed, $f_\ell^E(x)$ is of degree $O(\ell^2)$. In practice one cannot hope to compute $t \bmod \ell$ in this way for $\ell > 31$, say."
>
> Blake, Seroussi, and Smart, "Elliptic curves in cryptography," London Mathematical Society (1999), pages 111–112: The benefit of fast multiplication in Schoof's original algorithm is "mostly theoretical, and hard to realize in practical implementations"; the algorithm "will generally not suffice for the parameter ranges of practical interest."
>
> Vanstone, 5 July 2004 talk discussing Schoof's algorithm in the 1990s, according to notes by Bernstein: "For cryptographically interesting curves it just couldn't be used."

As a consequence, it is widely believed that point counting became practical only after Schoof's algorithm was improved by Elkies and Atkin.

My poster will show that Schoof's original algorithm is, in fact, practical for use with cryptographic-sized elliptic curves. For example, my implementation of Schoof's original algorithm (without improvements due to Elkies, Atkin, or Baby-Step Giant-Step) computes $\#E(F_p)$ in 1259 seconds (on a 2000MHz Athlon 64 X2) for a 160-bit prime p, and in 9577 seconds for a 256-bit prime p. The 160-bit computation could have been performed in under a day on a computer available in 1985. For comparison, Magma's implementation of the Schoof-Elkies-Atkin algorithm was reported in 2004 to take 2200 seconds (on an unspecified CPU) for a 400-bit prime p. Certainly the Elkies and Atkin improvements are valuable, but these improvements are not as drastic as is widely believed.

The speed of Schoof's original algorithm is of interest not only for historical reasons, but also for other applications of the underlying computational techniques, such as counting points in the Jacobian of a genus-2 hyperelliptic curve.

Date: April 28, 2006. Thanks to my advisor Daniel J. Bernstein for his guidance.