Math 168A Final Project: Computing a_p for Elliptic Curves

Neal Harris
Professor William Stein
UCSD

December 12, 2005

1 Introduction

In this paper, we examine some algorithms for computing a_p for a given elliptic curve E, and a prime number p, where:

$$\#E(\mathbb{F}_p) = p + 1 - a_p.$$

It turns out that computing a_p is crucial for computing the L-function $L(E, s)$ of an elliptic curve. We take this as sufficient motivation for computing a_p.

It is known that for an elliptic curve defined by:

$$y^2 = x^3 + ax + b$$

that:

$$a_p = -\sum_{x \in \mathbb{F}_p} \left(\frac{x^3 + ax + b}{p} \right). \tag{1}$$

This gives an $O(p^{1+o(1)})$ running time algorithm. But we can do better than this naïve approach. A more efficient way to compute a_p involves using the Baby-Step Giant-Step algorithm.

In this paper, we describe this algorithm (given in [1]), and give some examples of how it works.

2 The Algorithms

2.1 Hasse's Theorem

First, we state a useful theorem.

Hasse's Theorem. For any elliptic curve E defined over some finite field $\mathbb{F}_q$:

$$|\#E(\mathbb{F}_q) - (q+1)| \leq 2\sqrt{q}.$$

In particular, when $q = p$, we have:

$$|a_p| \leq 2\sqrt{p}.$$

2.2 Shanks' Baby-Step Giant Step Algorithm

The following is a useful method for solving the discrete logarithm problem. Let G be a group.

Baby-Step Giant Step. If we know:

$$\frac{B}{2} < C \leq |G| \leq B$$

then we can find $|G|$ in the following way:

1. Initialize. Set $h \leftarrow 1, C_1 \leftarrow C, B_1 \leftarrow B, S \leftarrow \{1\}, L \leftarrow \{1\}$.

2. Choose a random $g \in G$. Set $q \leftarrow \lceil \sqrt{B_1 - C_1} \rceil$.

3. Baby steps. Set $x_0 \leftarrow 1, x_1 \leftarrow g^h$. If $x_1 = 1$, then set $n \leftarrow 1$ and go to step 6. Otherwise, for each $2 \leq r \leq q-1$, set $x_r \leftarrow x_1 \cdot x_{r-1}$. For each $0 \leq r < q$, set $S_{1,r} \leftarrow x_r \cdot S, S_1 \leftarrow \bigcup_{0 \leq r < q} S_{1,r}$. If we find $1 \in S_{1,r}$, for $r > 0$, set $n \leftarrow r$ (for the smallest such r) and go to step 6. Otherwise, set $y \leftarrow x_1 \cdot x_{q-1}, z \leftarrow x_1^{C_1}, n \leftarrow C_1$.

4. Giant Steps. For each $w \in L$, set $z_1 \leftarrow z \cdot w$. Look for z_1 in S_1. If z_1 is found with $z_1 \in S_{1,r}$, set $n \leftarrow (n - r)$ and go to step 6.

5. Set $z \leftarrow y \cdot z, n \leftarrow (n + q)$. If $n \leq B_1$, go to step 4. Otherwise we have $|G| > B$. So we terminate the algorithm with an error message.

6. Set $n \leftarrow nm$.

7. For each prime p dividing n: (a) set $S_1 \leftarrow g^{n/q} \cdot S$; (b) if we have $z \in L$ such that $z \in S_1$, set $n \leftarrow n/p$ and go to step 7.

8. Set $h \leftarrow hn$. If $h \geq C$ then output h and terminate. In this case $|G| = h$. Otherwise, set $B_1 \leftarrow \lfloor B_1/n \rfloor, C_1 \leftarrow \lceil C_1/n \rceil, q \leftarrow \lceil \sqrt{n} \rceil, S \leftarrow \bigcup_{0 \leq r < q} g^r \cdot S, y \leftarrow g^q$, $L \leftarrow \bigcup_{0 \leq a \leq q} y^a \cdot L$, then go to step 2.

Now we can apply this algorithm with $G = E(\mathbb{F}_p), C = p + 1 - 2\sqrt{p}, B = p + 1 + 2\sqrt{p}$. This gives us an algorithm for computing a_p in $O(p^{1/4+o(1)})$ time.

But we can do even better.

2.3 The Shanks-Mestre Algorithm

We begin with a theorem:
Theorem. For an elliptic curve E defined by the following:

$$E : y^2 = x^3 + ad^2x + bd^3, d \neq 0$$

there are two isomorphism classes for all values of d. If we have $(\frac{d}{p}) = 1$, then the curve is isomorphic to the curve defined above with $d = 1$. For $(\frac{d}{p}) = -1$, these curves are isomorphic to another curve.

We state another theorem.
Theorem. Suppose we have two elliptic curves, E, and E', where:

$$\begin{aligned} E : y^2 &= x^3 + ad^2x + bd^3 \\ E' : y^2 &= x^3 + ae^2x + be^3 \end{aligned}$$

with $(\frac{d}{p}) = 1, (\frac{e}{p}) = -1$. Further, suppose the group structures are as follows:

$$\begin{aligned} E(\mathbb{F}_p) &\cong \mathbb{Z}/d_1\mathbb{Z} \times \mathbb{Z}/d_2\mathbb{Z} \\ E'(\mathbb{F}_p) &\cong \mathbb{Z}/d_1'\mathbb{Z} \times \mathbb{Z}/d_2'\mathbb{Z} \end{aligned}$$

with $d_1|d_2$, and $d_1'|d_2'$. Then for $p > 457$:

$$\max(d_2, d_2') > 4\sqrt{p}.$$

Armed with this result, we can state the Shanks-Mestre Algorithm:

1. Initialize. Set $x \leftarrow -1, A \leftarrow 0, B \leftarrow 1, k_1 \leftarrow 0$.
2. Repeat $x \leftarrow x + 1, d \leftarrow x^3 + ax + b, k \leftarrow (\frac{d}{p})$ until $k \neq 0$, and $k \neq k_1$. Set $k_1 \leftarrow k$. If $k = -1$, set $A_1 \leftarrow 2p + 2 - A \mod B$. Otherwise, set $A_1 \leftarrow A$.
3. Let m be the smallest integer such that $m > p + 1 - 2\sqrt{p}$ and $m \equiv A_1 \mod B$. Now use Baby-step Giant-step to find n such that $m \leq n < p + 1 + 2\sqrt{p}, n \equiv m \mod B$ and such that $n \cdot (xd, d^2) = 0$ on the curve $Y^2 = X^3 + ad^2X + bd^3$.
4. Factor n, and from this deduce the order h of (xd, d^2).
5. Find the smallest integer h' which is a multiple of h, and such that $h' \equiv A_1 \mod B$. If $h' < 4\sqrt{p}$, set $B \leftarrow \text{lcm}(B, h)$, and $A \leftarrow h' \mod B$ if $k_1 = 1$,
$A \leftarrow 2p + 2 - h' \mod B$ if $k_1 = -1$, then go to step 2.
6. Let N be the unique multiple of h' such that $p+1-2\sqrt{p} < N < p+1+2\sqrt{p}$. Output $a_p = p + 1 - k_1N$. Terminate.

This algorithm will run in $O(p^{1/4+\epsilon})$ time for any $\epsilon > 0$.

3 Examples

3.1 The Naïve Algorithm

Consider the curve E defined over $\mathbb{F}_7$:

$$E : y^2 = x^3 + 4x.$$

With such a small field, the naïve algorithm described in (1) gives a very reasonable way to compute a_p.

We have:

$$\begin{aligned} a_7 &= -\Big\{\Big(\frac{0}{7}\Big) + \Big(\frac{5}{7}\Big) + \Big(\frac{2}{7}\Big) + \Big(\frac{4}{7}\Big) + \Big(\frac{3}{7}\Big) + \Big(\frac{5}{7}\Big) + \Big(\frac{2}{7}\Big)\Big\} \\ &= 0 - 1 + 1 + 1 - 1 - 1 + 1 \\ &= 0. \end{aligned}$$

Now we check this with SAGE:

```
sage: E = EllipticCurve([4,0])
sage: E
Elliptic Curve defined by y^2  = x^3 + 4*x over Rational Field
sage: E.ap(7)
0
```

3.2 Shanks-Mestre

Now, we show explicitly how the Shanks-Mestre Algorithm works. We will suppress some of the details in the computation (i.e. use SAGE to compute multiples of points on E). Consider the following elliptic curve defined over $\mathbb{F}_{499}$:

$$E : y^2 = x^3 + x.$$

Step 1 is rather easy. We set $x \leftarrow -1, A \leftarrow 0, B \leftarrow 1, k_1 \leftarrow 0$. Now, in step 2, we increment x so that we have $x = 0$. But then we have $d = x^3 + x = 0$, and therefore $(\frac{d}{p}) = 0$. So, we repeat. Now we have $x = 1 \implies d = 2 \implies (\frac{d}{p}) = -1$. Since we have $B = 1$, we needn't change A_1.

Now we see that since $B = 1$, and $455 < p + 1 - 2\sqrt{p} < 456 \implies m = 456$. Now, using SAGE, we have the following:

```
sage: d = 2
sage: p = 499
sage: E = EllipticCurve(GF(p), [d^2, 0])
sage: E
```

```
Elliptic Curve defined by y^2  = x^3 + 4*x over Finite field of size 499
sage: x = 1
sage: P = E([x*d, d^2])
sage: 456*P
0
```

So in fact our n is $n = 456$. Now, factoring gives $n = 2^3 * 3 * 19$. Now, we begin looking for the order of P.

```
sage: 2*P
(0, 0)
sage: 4*P
0
```

So we see that $|P| = 4$. Now $h \leftarrow 4$. Again, since $B = 1$, we have $h' = 4$ as well. Since $4 < 4\sqrt{p}$, we set $B \leftarrow \mathrm{lcm}(B, h) = \mathrm{lcm}(1, 4) = 4$. Now, we have $2p + 2 - h' = 996$, which is a multiple of 4. So, we still have $A = 0$. Now, we return to step 2.

We increment x, and have $x = 2$. So $d = 10 \implies (\frac{d}{p}) = -1$. So we repeat. Now we have $x = 3$. This gives $d = 30 \implies (\frac{30}{p}) = 1$. So we have have $k \leftarrow 1$, and set $k_1 \leftarrow 1$. And we already have $A = A_1$.

Since $4|456$, and $A_1 = 0$, we still have $m = 456$. Also, we know that $544 < p+1+2\sqrt{p} < 545$. Now, using SAGE, we have:

```
sage: E = EllipticCurve(GF(499), [900, 0])
sage: P = E([90, 900])
sage: for i in range(456, 544):
   ....:    if (i*P==0):
   ....:        i;
   ....:
_31 = 500
```

So we have $n = 500$. Factoring, we have $n = 2^2 5^3$. Now, let's find the order of P:

```
sage: 250*P
(0, 0)
sage: 100*P
0
sage: 20*P
(213, 394)
sage: 4*P
(405, 201)
```

So, we see that $|P| = 100$. With this, we set $h \leftarrow 100$. Since $B = 4$, we have $B|h$, and therefore $h \equiv A_1 \mod B$. So, set $h' \leftarrow h = 100$. Note that $h' > 4\sqrt{p}$.

Finally, we see that the unique multiple of h' such that $p+1-2\sqrt{p} < N < p+1+2\sqrt{p}$ is $5 \cdot 100 = 500$. So, set $N \leftarrow 500$.

And now we output $a_p = p + 1 - k_1 N = 499 + 1 - 1 \cdot 500 = 0$.

Let's check this with SAGE:

```
sage: E = EllipticCurve([1,0])
sage: E
Elliptic Curve defined by y^2  = x^3 + x over Rational Field
sage: E.ap(499)
0
```

Bingo.

4 Computing a_p for Large p

There are many situations in which we would like to calculate a_p for very large p; cryptography is the canonical example of such a situation. It turns out that there is a better algorithm for computing a_p for very large primes. Let q be a large prime. Let E_q be an elliptic curve over $\mathbb{F}_q$.

This algorithm is known as Schoof's Algorithm, after Rene Schoof. First we recall that:

$$-2\sqrt{q} < a_q < 2\sqrt{q}. \tag{2}$$

Pick some collection of smaller primes $p_1, p_2, \ldots, p_k$ such that:

$$p_1 p_2 \ldots p_k > 4\sqrt{q}.$$

Then, compute $a_q \mod p_i$ for $1 \leq i \leq k$. Use the Chinese Remainder Theorem to compute $a_q \mod p_1 p_2 \ldots p_k$. But then we compute $a_q \mod 4\sqrt{q}$, and by 2 above, we can compute a_q exactly.

This algorithm does have an asymptotically better running time than Shanks-Mestre. Specifically, the running time is $O\big(\ln^8(q)\big)$. It is important to note that Schoof's algorithm is only *asymptotically* better. For medium sized primes (approximately for $p < 2^{60}$), Shanks-Mestre is still faster.

We suppress any further details (including how exactly to compute $a_q \mod p_i$ for the various p_i). Instead, we reference [2].

5 Some Data

In what follows, we give some timing information for computing a_p for various primes, and for the following curve E:

$$E : y^2 = x^3 + x + 1.$$

For those readers familiar with the notion of complex multiplication on an elliptic curve, note that E does not have complex multiplication. If it did, computing a_p would be much easier. This is because there is an algorithm which is beyond the scope of this paper that computes a_p much faster for curves that are equipped with complex multiplication than for those curves that do not have complex multiplication.

What follows is a table of timing information. All computations were performed on a dual Opteron 248 Sun Fire V20Z server with 8GB RAM. The software package used is SAGE.

p	a_p	CPU time (s)
1000000000000037	1847783	0.01
100000000000000000039	6324941747	0.38
10000000000000000000009	139275907750	1.88
100934583920633341444919	-72259137428	2.58
10000000000000000000000013	3919779458826	13.48
1000000000000000000000000103	-44679400742701	49.03
10000000000000000000000000331	-38606803965466	60.17
11000000000000000000000000117	-94320506755356	77.94
50000000000000000000000000143	141508045851704	118.81
97000000000000600000000000031	214203597842946	150.07
99900048574389597849375783563	13989829642156	109.45
99904857484389597849375783601	-375287338085352	177.12
1000000000000000000000000000057	1911205794915458	396.16
7063271223590103947858054109143	2019116948430037	512.01

To compute a_p for all primes $p < 10^6$ took 9.18 seconds.

6 Concluding Remarks

Computing a_p for an elliptic curve E gives us information about the L-function $L(E, s)$ for E. Since we can learn a lot from L, computing a_p is worthwhile.

By using the Hasse bound and the Baby-Step Giant-Step Algorithm, we can efficiently compute a_p for $p > 457$ with the Shanks-Mestre Algorithm.

However, Shanks-Mestre is too slow if we wish to use very large primes, as in more than 60 bits, say. In cases with large primes, it is more efficient to use Schoof's Algorithm to compute a_p. This method uses information about a_p modulo a collection of smaller primes to infer a_p.

References

[1] H. Cohen. *A Course in Computational Algebraic Number Theory.* Springer, 1996.

[2] R. Schoof. Counting points on elliptic curves over finite fields, 1995.

[3] W. Stein and D. Joyner. Sage: System for algebra and geometry experimentation, 2005.