Computing l-isogenies with the p-torsion

Jean-Marc Couveignes[*1]

Universiteit Utrecht

Abstract. Computing l-isogenies between elliptic curves defined over a finite field $\mathbb{F}_q$ of small characteristic is of some importance for the computation of the cardinalities of elliptic curves using Schoof-Atkin-Elkies method. In a previous publication [3] we showed that this could be achieved in $O(l^{3+\epsilon})$ multiplications in $\mathbb{F}_q$ using formal groups. This method has been implemented by Lercier and Morain who obtained spectacular results in this direction [7, 8]. Nevertheless, the use of formal groups seems to be a serious deterent both for man and machine to perform such a work. More recently Lercier proposed an algorithm specific for characteristic 2 that has the same assymptotic complexity but is faster by some significative constant factor. In this paper we propose a general algorithm which does not use formal groups. Instead we take advantage of the elementary Galois properties of the p-torsion. This algorithm has the same complexity as the previous ones if we don't use fast multiplication techniques. But, contrary to the previous methods, it allows the use of fast multiplication for polynomials and then turns out to run in $O(l^{2+\epsilon})$ multiplications in the field $\mathbb{F}_q$. Our algorithm has also the advantage that it is made exclusively of very classical routines in polynomial and elliptic curve arithmetic. Also one may expect that the implementation of this method should require less work than the previous ones thus bringing new people to this kind of calculation.

1 Introduction

The first polynomial time algorithm for the enumeration of points on elliptic curves defined over finite fields was given by Schoof in [11]. This algorithm was made efficient by Atkin and Elkies [12, 10, 4, 1, 2]. Elkies improvement requires the computation of some explicit l-isogeny between two given elliptic curves known to be l-isogenous. Elkies proposed nice modular equations for this problem. This is quite efficient as long as $l < p$ where p is the characteritic. For the case $p < l$ we proposed in [3] an algorithm based on formal groups. This was quite successfuly implemented by Lercier and Morain [7, 8]. Recently Lercier [6] discovered surprising formulae, valid in the case of characteristic 2, that lead to an algorithm faster than our original one by some constant. This seems to be due to the apparition of many linear equations in the presentation of the problem. We don't know whether this method is likely to be generalized to the case of arbitrary characteristic. Here we propose a new algorithm for the computation

[*] Membre de l'Option Recherche du Corps des Ingénieurs de l'Armement

of isogenies in small characteristic that avoids the use of formal groups. It works in any characteristic and it uses only simple and classical computational routines such as factoring polynomials over finite fields. Its complexity is better since it computes an isogeny of degree l in time $O(l^{2+\epsilon})$ operations in $\mathbb{F}_q$ provide we use fast multiplication techniques and allow a precomputation which takes time $O(l^{3+\epsilon})$ operations in $\mathbb{F}_q$ and is useful for all isogenies of degree at most l. In the context of Schoof-Atkin-Elkies's algorithm this means that we will compute all the isogenies we need in time $O(l^{3+\epsilon})$ operations in $\mathbb{F}_q$ that is assymtotically negligible compared to the rest of the algorithm.

We assume all along this paper that we are given a finite field $\mathbb{F}_q$ with $q = p^e$ and two elliptic curves E_a and E_b defined over $\mathbb{F}_q$ and l-isogenous over $\mathbb{F}_q$ for l a prime different from p. We assume that the two curves are non-supersingular. To make the presentation easier we assume that $p > 3$. We want to compute an explicit l-isogeny between those curves. We use the fact that such an isogeny I must send the p^k-torsion of E_a onto the p^k-torsion of E_b. To make it simple assume that all the p^k-torsion points of E_a and E_b are rational over $\mathbb{F}_q$. Let P_a be some p^k-torsion point on E_a. For any p^k-torsion point P_b on E_b let us suppose that $I(P_a) = P_b$. Then for any integer $0 \leq m < p^k$ we have $I([m]P_a) = [m]P_b$ and if $p^k > \sqrt{2}l$ we completely characterize I. In particular we can check whether our assumption that $I(P_a) = P_b$ was correct. If it is not, we try another P_b.

Let us see more in detail how the knowledge of the images of the $[m]P_a$ gives an algebraic description of I. We use the following notation. If a point P has x-coordinate x_P then the x-coordinate of $[m]P$ is denoted by $x_{[m]P}$ or simply $[m]x$. Therefore $[m]x$ is a rational fraction in x of degree m^2. First it is evident that two isogenies of degree l that agree on the $[m]P_a$'s are identical. Indeed, the difference is of degree lower than $\sqrt{2}l$ because of the triangle inequality and since it cancels over the p^k-torsion and $p^k > \sqrt{2}l$ we are done.

Let $J(x) = u(x)/v^2(x)$ be the rational fraction such that the image of a point $P_a = (x_a, y_a)$ by I is $I(P_a) = P_b = (x_b, y_b)$ with $x_b = J(x_a)$. We have $\deg(v) = (l-1)/2$ and $\deg(u) = l$. Since we know the x-coordinates of the p^k-torsion points and their images by J, we can find by interpolation a polynomial $A(x)$ of degree smaller than or equal to $(p^k - 3)/2$ such that $A([m]x_a) = [m]x_b = J([m]x_a)$ for all m's. We then have

$$A(x) = u(x)/v^2(x) \bmod T_{a,k}(x)$$

where $T_{a,k}(x)$ is the p^k-torsion polynomial of E_a.

Then if p^k is bigger enough than $4.l$ this is enough to compute $u(x)$ and $v(x)$. Indeed from the above congruence we know that there exists a polynomial $B(x)$ of degree smaller than or equal to $l-2$ such that

$$u - Av^2 = T_{a,k}B.$$

Now if we note $\hat{B}(x) = B(1/x)x^{l-2}$ and $\hat{A}(x) = A(1/x)x^{\frac{p^k-3}{2}}$ and $\hat{v}(x) = v(1/x)x^{\frac{l-1}{2}}$ and $\hat{T}_{a,k}(x) = T_{a,k}(1/x)x^{\frac{p^k-1}{2}}$ we get

$$\frac{\hat{A}}{\hat{T}_{a,k}} + \frac{\hat{B}}{(\hat{v})^2} = x^{\frac{p^k-5}{2}} \frac{\hat{u}}{(\hat{v})^2 \hat{T}_{a,k}} = O(x^{\frac{p^k-5}{2}})$$

and this series identity is enough to compute B and v using continued fractions.

Of course the p^k-torsion points are not defined over $\mathbb{F}_q$ in general and this makes the interpolation step a bit more tricky as we shall see in the next section.

2 More details

We give here more details on the algorithm and its complexity.

In the rest of this paper we restrict our attention to *primitive* torsion points. Therefore p^k-torsion will mean the set of all $p^{k-1}(p-1)$ torsion points of exact order p^k.

2.1 Complexity

We recall that the product, quotient , gcd and inverse modulo of two polynomials of degree d require $O(d^2)$ multiplications in the field if we don't use any fancy method. If we use fast multiplication techniques we achieve all these in time essentially linear.

We will also need to compute the roots of a squarefree splitting polynomial $F(x)$ of degree d over an extension field $\mathbb{K}$ of $\mathbb{F}_p$ of degree h. This can be done in time $O(hd^4)$ multiplications in $\mathbb{K}$ using the algorithm for small characteristic given in [9] chapter 4, section 3. Note that the constant depends on p.

To find a non trivial factor of F, this algorithm requires the computation of

$$\gcd(F(x), S(\beta^j x) - c)$$

where

$$S(x) = \sum_{i=0}^{h-1} x^{p^i}$$

and β is any given generator of the field $\mathbb{K}$ over $\mathbb{F}_p$ and c takes all the values in $\mathbb{F}_p$ in turn and $0 \leq j \leq h-1$. If we compute all these ph gcds we are sure to break F in at least two pieces. This is because the polynomial is squarefree and the trace form is non degenerate. In general, taking $j = 1$ is enough.

We start computing $S(x) \bmod F(X)$ which requires $O(hd^2)$ multiplications in $\mathbb{K}$. Then for a given β^j the computation of $S(\beta^j x) \bmod F(x)$ requires $O(d^3)$ multiplications. And the computations of all gcds requires $O(pd^2)$ multiplications. Since we do that for all j's in the worst cases we have $O(hd^3)$ mutiplications for a breaking and therefore $O(hd^4)$ for the whole splitting.

2.2 Description of the p^k-torsion

The action of the Frobenius Φ on the p^k-torsion for $p^k < q$ is just multiplication by the trace t. Indeed it is multiplication by some prime to p integer λ (otherwise some power of the Frobenius would have the p^k-torsion in its kernel) satisfying $\lambda(\lambda - t) = 0 \pmod{p^k}$. Also this trace t is the same on any two isogenous curves.

Therefore the p^k-torsion polynomials of E_a and E_b each factor over $\mathbb{F}_q$ in f factors of degree d with $df = p^{k-1}(p-1)/2$.

Let $1 = m_1, m_2, ..., m_f$ be integers in the interval $[1, p^k[$ that form a system of coset representatives of the subgroup generated by t in $(\mathbb{Z}/p^k\mathbb{Z})^*$. Then $T_{a,k}$ factors over $\mathbb{F}_q$ as

$$T_{a,k} = \prod_{i=1}^{f} U_{a,i}$$

where the $U_{a,i}$'s are of degree d and they are named in such a way that if we call $x_a = x \bmod U_{a,1}(x)$ one fixed root of $U_{a,1}$ then $[m_i]x_a$ is a root of $U_{a,i}$.

We also call $(n_i)_i$ a set of representatives of the inverses of the m_i. It is crucial that these numbers should not be changed along the algorithm because they provide a coherent system of isomorphisms between the fields $\mathbb{F}_q[x]/U_{a,i}(x)$. Indeed let us call $x_{a,i} = [m_i]x_a$. Then let us define $A_{i,j} = [m_j][n_i](x) \bmod U_{a,i}$. Then we have $A_{i,j}(x_{a,i}) = x_{a,j}$.

Of course, there is a similar factorization of $T_{b,k}$ and the rational fraction J maps the factors of $T_{a,k}$ onto the factors of $T_{b,k}$ but we do not know how because we still ignore the image x_b of x_a by J.

Although we do not know x_b we can at least compute a root γ of $T_{b,k}(x)$ over $\mathbb{K} = \mathbb{F}_q[x]/U_{a,1} = \mathbb{F}_q(x_a)$. This can be done efficiently by *successively* factoring k-polynomials of degree p instead of one of degree p^k.

Call σ the little Frobenius, i.e. $\sigma(u) = u^p$ for $u \in \mathbb{F}_q$. It is an automorphism of $\mathbb{F}_q$ and $\sigma^{-1} = \sigma^{e-1}$. We call $E_{b,1}$ the curve obtained from E_b by conjugating the coefficients by σ^{-1}. Let us call ϕ the purely inseparable isogeny from $E_{b,1}$ to E_b that maps (x, y) to (x^p, y^p). The dual of ϕ is called V. It is the separable isogeny of degree p obtained by quotienting E_b by its p-torsion.

$$\begin{tikzcd}
E_b \arrow[d, bend right=60, "V"'] \\
{}^{\sigma^{-1}}E_b \arrow[u, bend right=60, "\phi"']
\end{tikzcd}$$

Now if we want to compute a p^2-torsion point on E_b we start by computing a p-torsion point on $E_{b,1}$ and then we look for some point on E_b which is maped onto this point by V. If we need p^k-torsion points we must start with some p-torsion point on $E_{b,k-1} = {}^{\sigma^{-k+1}}E_b$ and then compute preimages by all the successive separable isogenies of degree p.

The cost of this computation is the cost of k splittings of separable polynomials of degree p over an extension of degree d of $\mathbb{F}_q$, i.e. an extension of degree

$h = ed$ of $\mathbb{F}_p$. The complexity is therefore $O(khp^4)$ multiplications in this field. We end up with a complexity of $O(\log(q)l\log(l))$ multiplications in a field of degree h. To do this efficiently we must use a fast multiplication algorithm. This will be efficient since h is typically of size 10^5. Then we will have a complexity of $O((\log(q)l)^{2+\epsilon})$ operations in $\mathbb{F}_p$ the primitive field.

But we can do much better. Indeed, if we use the Hasse function of the elliptic curve, the computation of the preimage of some point by the V isogeny boils down to solving an equation of the form $x^p - x - C = 0$ where C is some element in $\mathbb{F}_q[x]/U_{a,1}$. For explicit expressions relating the Hasse functions and the Weierstrass ones see [5, 13]. This is an affine polynomial [9] and can be solved in $O(l^2)$ multiplications in $\mathbb{F}_q$ as we explain in section 2.4. This is indeed much more efficient.

We now have the x-coordinate γ of some p^k-torsion point on E_b. In the following we will assume that $\gamma = x_b = J(x_a)$ i.e. we have found the *right* torsion point. If this is not true the method of the next section will fail and we then have to replace γ by some multiple of it until it works.

2.3 Interpolation

We here assume that we know some p^k-torsion point $P_a = (x_a, y_a)$ on E_a and its image $P_b = (x_b, y_b)$ by I. This means that $x_a = x \bmod U_{a,1}(x)$ and $x_b = \Delta(x) \bmod U_{a,1}(x)$ where Δ is a polynomial of degree $d-1$.

Now if α is any root of $U_{a,1}$ then $J(\alpha) = \Delta(\alpha)$. This is because J is defined over $\mathbb{F}_q$. Therefore the polynomial $A(x)$ we are looking for is congruent to Δ modulo $U_{a,1}$. In the same manner we define he polynomial Δ_i as $\Delta_i = [m_i]_b\Delta[n_i]_a \bmod U_{a,i}$ where $[n_i]_b$ is the rational fraction for the multiplication by n_i on E_b and $[m_i]_a$ is the multiplication by m_i on E_a.

Then we have

$$A(x) = \Delta_i \pmod{U_{a,i}}$$

and we finish using chinese remainder theorem.

2.4 Linear polynomials over composite extensions

It is known that linear polynomials are easy to factor, see [9] and this has been used in [7].

Here, we have a slightly more subtle situation because the field $\mathbb{K}$ over which we want to factor is an extension of $\mathbb{F}_q$ of degree d let's say. And $q = p^e$. Since the polynomial $x^p - x$ is a linear map over $\mathbb{F}_p$ we may compute the matrix and inverse it over its image in time $O((ed)^3)$ operations in $\mathbb{F}_p$ but this is too much for us. Instead, we consider the polynomial map $x^q - x$ that is linear over $\mathbb{F}_q$ and inverse the corresponding matrix in time $O(d^3)$ multiplications in $\mathbb{F}_q$. We also inverse the operator $x^p - x$ over $\mathbb{F}_q$ in time $O(e^3)$ operations in $\mathbb{F}_p$. In the situation that we consider all this precomputation is done in time $O(l^3)$

multiplications in $\mathbb{F}_q$ but we notice that it will be useful for the computations of *any* isogeny from E_a. Since we have to compute $O(l)$ of them, it is worth.

Now if we have to solve some equation $x^p - x = C$ we start computing the "trace" $c = C + C^p + C^{p^2} + \ldots + C^{p^{e-1}}$ of C in the extension $\mathbb{F}_q/\mathbb{F}_p$ although C is not in $\mathbb{F}_q$!. Then we solve $x^q - x = c$ using the $\mathbb{F}_q$-linearity of the operator $x^q - x$, in time $O(l^2)$ multiplications in $\mathbb{F}_q$ because we already inversed this operator. We then obtain a solution x_0 to this equation. All solutions are of the form $x_0 + \lambda$ with $\lambda \in \mathbb{F}_q$. We then look for some λ such that $(x_0 + \lambda)^p - x_0 - \lambda = C$. This is just solving $\lambda^p - \lambda = C + x_0 - x_0^p$ in $\mathbb{F}_q$ and $C + x_0 - x_0^p$ is in $\mathbb{F}_q$ (exercice). This is done in e^2 multiplications in $\mathbb{F}_p$ i.e. nothing.

3 Conclusion

We have given another algorithm for the computation of isogenies in small characteristic. We expect it to be useful to people willing to work efficiently on elliptic curve using basic routines for finite fields. This new method allows a better use of fast multiplication techniques. We can first precompute in time $O(l^{3+\epsilon})$ multiplications the data that are useful for the computations of all isogenies of degree smaller than l.

Then each isogeny of degree l requires $O(l^{2+\epsilon})$ multiplications in $\mathbb{F}_q$.

Aknowledgements. The author wishes to thank R. Lercier and F. Morain for many enlightening discussions and a long collaboration in this field.

Neither this work nor any other work would have been possible without the providential and fruitful hospitality of the University of Utrecht.

References

1. A. O. L. Atkin. The number of points on an elliptic curve modulo a prime. Preprint, 1988.
2. A. O. L. Atkin. The number of points on an elliptic curve modulo a prime (ii). Preprint, 1992.
3. J.-M. Couveignes. *Quelques calculs en théorie des nombres.* Université de Bordeaux, 1994.
4. N.D. Elkies. Explicit isogenies. 1991.
5. Hiroshi Gunji. The hasse invariant and p-division points of an elliptic curve. *Arch. Math. (Basel)*, 27:148–158, 1976.
6. R. Lercier. Computing isogenies in characteristic 2. Submitted for publication at ANTS 2.
7. R. Lercier and F. Morain. Counting points on elliptic curves over F_{p^n} using Couveignes's algorithm. Research Report LIX/RR/95/09, École Polytechnique–LIX, September 1995.
8. R. Lercier and F. Morain. Counting the number of points on elliptic curves over finite fields: strategies and performances. In L.C. Guillou and J.-J. Quisquater, editors, *Advances in cryptology, EUROCRYPT 95*, volume 921 of *Lecture notes in computer science*, pages 79–94. Springer, 1995.

9. R. Lidl and H. Niederreiter. *Introduction to finite fields and their applications.* Cambridge University Press, 1986.
10. François Morain. Calcul du nombre de points sur une courbe elliptique dans un corps fini : aspects algorithmiques. Submitted for publication of the Actes des Journées Arithmétiques 1993, March 1994.
11. R. Schoof. Elliptic curves over finite fields and the computation of square roots mod p. *Math. of Comp.*, 44:483–494, 1985.
12. René Schoof. Counting points on elliptic curves over finite fields. to appear in the Journal de Théorie des nombres de Bordeaux.
13. J. F. Voloch. Explicit p-descent in characteristic p. *Comp. Math.*, 74:247–258, 1990.

This article was processed using the LaTeX macro package with LLNCS style