Some remarks on the elliptic curve discrete logarithm

P. Gaudry

November 25, 2003

1 Introduction

The elliptic curve discrete logarithm problem is the basis of numerous cryptosystems. We give an algorithm to efficiently lift an instance of a discrete log problem to a p-adic field. We then explain an attempt to use this tool to break the discrete log, and we try to make clear the point that makes it fail.

Attached to this note, there is some piece of Magma code that illustrate our algorithm.

2 Lifting of ECDLP to a p-adic field

2.1 The main result

Let E be an elliptic curve over a finite field $\mathbb{F}_q$ with q elements. Let P and Q be two points on E such that $Q = \lambda P$, and let n be the order of P. Of course, the discrete logarithm λ is supposed to be unknown. We also assume that n is coprime to the characteristic p of $\mathbb{F}_q$; indeed there exist algorithms to compute efficiently the p-part of the discrete log [6, 3, 4, 2].

Let K be the unramified p-adic field with residue class field $\mathbb{F}_q$. Let $\mathcal{E}$ be any lift of E to K. It can be a canonical lift, but this is not required. The goal of this section is to show that it is possible to construct points $\mathcal{P}$ and $\mathcal{Q}$ on $\mathcal{E}$ that reduce to P and Q modulo p, and that are such that $\mathcal{Q} = \lambda\mathcal{P}$. More precisely, we are going to show that such $\mathcal{P}$ and $\mathcal{Q}$ can be constructed to precision k in $O(\mathsf{M}(k \log q) \log n)$ bit-operations, where M is a function such that two integers of d bits can be multiplied in time $\mathsf{M}(d)$.

The strategy is to use the n-th divsion polynomial ψ_n, and to construct the abscissae of $\mathcal{P}$ and $\mathcal{Q}$ as lifted roots of ψ_n. We want to use the Newton method and therefore we need to be able to evaluate ψ_n and its derivative efficiently. This is the purpose of the next two lemmas.

Lemma 1 *Let ψ_n be the n-th division polynomial of an elliptic curve defined over a field K. For any x in K, the value $\psi_n(x)$ can be computed in $O(\log n)$ operations in K.*

Proof. The polynomial ψ_n is related to other division polynomials of order about $n/2$ by well known recurrence formulae that can be found for instance in [1]. Following these recurrence formulae yields a recursive algorithm that computes $\psi_n(x)$. During the process, if we take care to remember the values of k for which the value of $\psi_k(x)$ has already been computed, then the number of operations is proportionnal to the number of k for which $\psi_k(x)$ is computed during the recursion. Since at each step the degree is roughly halved, there are $O(\log n)$ such k's, and the Lemma follows. □

Lemma 2 *Let ψ_n be the n-th division polynomial of an elliptic curve defined over a field K. For any x in K, the value $\psi'_n(x)$ can be computed in $O(\log n)$ operations in K.*

Proof. The recurrence formulae that relate division polynomials can be differentiated. Therefore, $\psi'_n(x)$ can be expressed in terms of division polynomials of order about $n/2$ and their derivatives, which translates in a recursive algorithm. The same argument as before stands: we have to evaluate the number of k for which $\psi_k(x)$ and $\psi'_k(x)$ have to be computed. And this number is again in $O(\log n)$. □

From those two lemmas, the Newton method for lifting the abscissa of a n-torsion point is deduced, as shown in Algorithm 1.

Algorithm 1 Lifting the abscissa of a n-torsion point

1. $x \leftarrow x$-coordinate of the point to lift, viewed as an element of K at precision 1.
2. $k \leftarrow 1$
3. While k is less than the wanted precision do
   (a) Lift x arbitrarily to precision $2k$;
   (b) $x \leftarrow x - \frac{\psi_n(x)}{\psi'_n(x)}$;
   (c) $k \leftarrow 2k$;
4. Return x;

Theorem 1 *Let E be a curve over $\mathbb{F}_q$, let P be a point of known order n in E, and Q a point of E such that there exists a λ with $Q = \lambda P$. Let $\mathcal{E}$ be curve which is a lift of E to a p-adic field K. Then it is possible to lift P and Q to points $\mathcal{P}$ and $\mathcal{Q}$ of $\mathcal{E}$ such that $\mathcal{Q} = \lambda\mathcal{P}$, to precision k, in $O(\mathsf{M}(k \log q) \log n)$ bit-operations.*

Proof. For computing $\mathcal{P}$, we start by computing its abscissa from the abscissa of P using Algorithm 1. Then its ordinate y can be obtained by another Newton lifting based on the equation of the curve. Thus we obtain a point $\mathcal{P}$ which is a n-torsion point of $\mathcal{E}$ and whose reduction modulo p is P. We do exactly the same procedure for computing $\mathcal{Q}$ from Q.

As n is coprime to p, the reduction modulo p is an isomorphism from the n-torsion points of $\mathcal{E}$ to the n-torsion points of E. Therefore, the relation $Q = \lambda P$ is transported into $\mathcal{Q} = \lambda\mathcal{P}$.

We now analyse the running time of this process. In Algorithm 1, each loop in Step 3 doubles the precision of the result. Indeed, we are in the typical case of the Newton method: by induction hypothesis, $\psi_n(x)$ has valuation at least k and since n is coprime to p, the polynomial ψ_n is separable modulo p and $\psi_n(x)$ has valuation 0. Therefore, after Step 3.(b), x is an approximation to precision $2k$ of a root of ψ_n. Hence the number of times this loop has to be done is $O(\log k)$, where k is the required precision. Due to the previous lemmas, one iteration of Step 3 costs $O(\log n)$ operations in K at the current precision. And finally, as usual in a Newton method, due to the fact that the computations at high precision are much more costly than those at low precision, it can be shown that the cost of the whole computation is a constant times the cost of the last iteration, at maximal precision. One operation in K at precision k can be done with $O(\mathsf{M}(k \log q))$ bit-operations. Therefore we obtain the announced complexity. □

2.2 Practical signifiance

In cryptography, ususally q is a prime or a prime power of size about 200 bits and n is a prime integer of about the same size. For a reasonable value of the precision k (say, up to a few hundreds) at which we want to lift the problem, the method described in Theorem 1 is then very efficient. On an average computer, with a non-optimized implementation using a Computer Algebra System or a Number Theory Library, this should be a matter of hours or even minutes, which is nothing in the context of an attack on the discrete log problem.

The formulae for the partial derivatives of ψ_n are tedious to write and instead of putting them in the proof of the lemma, we preferred to give some piece of code that the reader can run inside Magma or can convert to its favorite language. The formulae are in a file called `SLPs.mag` and an example `example.mag` demonstrates how they can be used.

2.3 Other ways to do the lifting

In the previous sections, we assumed that the lifted curve $\mathcal{E}$ was fixed and then we lifted the points $\mathcal{P}$ and $\mathcal{Q}$ accordingly. It is also possible to do the converse: given P and Q on a curve E over a finite field as before, we fix any lift of the abscissa of the point P. Then Algorithm 1 can be modified as follows: the variable that we are going to lift is no longer x but a coefficient A of the equation of the curve E. We start with the initial value of A, and its value is lifted using a Newton operator based on $\psi_n(x)$, viewed as a univariate polynomial in A (remember that here x is a fixed scalar in the p-adic field). Therefore the derivative that occurs in the denominator is the partial derivative of ψ_n with respect to A and can be computed efficiently. As a conclusion, we obtain the lifted parameter A and the corresponding curve $\mathcal{E}$ for which x is the abscissa of a n-torsion point $\mathcal{P}$ lifting P. The y-coordinate of $\mathcal{P}$ and the lifting of $\mathcal{Q}$ are done in the same way as in Theorem 1. Thus we have the following result:

Theorem 2 *Let E be a curve over $\mathbb{F}_q$, let P be a point of known order n in E, and Q a point of E such that there exists λ with $Q = \lambda P$. Let x be a lift to a p-adic field K of the abscissa of P. Then it possible to construct a curve $\mathcal{E}$ that lifts E to precision k, together with two points $\mathcal{P}$ and $\mathcal{Q}$ on $\mathcal{E}$ such that x is the abscissa of $\mathcal{P}$ and $\mathcal{Q} = \lambda\mathcal{P}$, in $O(\mathsf{M}(k \log q) \log n)$ bit-operations.*

In fact, the equation of an elliptic curve can be given by 2 parameters, for instance if $p \geq 5$, the equation can be of the form $y^2 = x^3 + Ax + B$. Therefore it is possible to fix both the abscissa and the ordinate of $\mathcal{P}$ and to lift A and B, so that $\mathcal{P}$ is an n-torsion point on the corresponding curve. This involve a multivariate Newton lifting, and is more technical, but there is no theoretical difficulty here and the running time is the same up to a constant factor.

Theorem 3 *Let E be a curve over $\mathbb{F}_q$, let P be a point of known order n in E, and Q a point of E such that there exists a λ with $Q = \lambda P$. Let $\mathcal{P}$ be a lift to a p-adic field K of P. Then it possible to construct a curve $\mathcal{E}$ that lifts E to precision k, together with a point $\mathcal{Q}$ on $\mathcal{E}$ such that $\mathcal{Q} = \lambda\mathcal{P}$, in $O(\mathsf{M}(k \log q) \log n)$ bit-operations.*

If we authorize more free parameters in the equation of the lifted curve, we could in principle also fix $\mathcal{P}$ and $\mathcal{Q}$ in advance.

3 An attack that cannot work, using the p-adic elliptic logarithm

It is tempting to try to use the p-adic elliptic logarithm, since it has proven to be deadly usefull in the context of anomalous curves. Let us see how it goes, and why it is not so simple. I am pretty sure that many people have thought about what is described here before me. Our goal is to show that our lifting algorithm does not change the situation.

The notations are the same as above: we have a curve $\mathcal{E}$ with points $\mathcal{P}$ and $\mathcal{Q}$ that lift a curve E with the points P or Q for which we want to solve the discrete log. For the moment, we assume nothing about $\mathcal{P}$ and $\mathcal{Q}$ except that they lift P and Q. In particular we do not ask that $\mathcal{Q} = \lambda\mathcal{P}$ (except that it has to be true modulo p).

Recall the classical notations of [5]: $\mathcal{E}_0$ is the group of points of $\mathcal{E}$ that can be reduced modulo p to a non-singular point (in our case, $\mathcal{E}_0$ is just $\mathcal{E}$), and $\mathcal{E}_1$ is the subgroup of $\mathcal{E}_0$ consisting of the points that reduce to the neutral (the point at infinity) on E. The group $\mathcal{E}_1$ can be seen as the points that are close to the neutral of $\mathcal{E}$ in the p-adic sense.

Then there is a classical result that states that $\mathcal{E}_0$ is isomorphic as a group to E times $\mathcal{E}_1$. Let π denote the projection that map an element of $\mathcal{E}_0$ to E by reducing the coordinates modulo p. Then the map can be explicitly given by

$$\begin{array}{rccl} \varphi & : & \mathcal{E}_0 & \to & E \times \mathcal{E}_1 \\ & & \mathcal{P} & \mapsto & (\pi(\mathcal{P}), \mathcal{P} - \langle \pi(\mathcal{P}) \rangle). \end{array}$$

To explain what we mean by the symbols $\langle\ \rangle$, we first insist on the fact that without them, the map φ is ill-defined. Indeed, the image $\pi(\mathcal{P})$ as a canonical sense only modulo p, whereas when we write $\mathcal{P} - \pi(\mathcal{P})$, we understand the difference (for the group law in $\mathcal{E}$) of $\mathcal{P}$ and $\pi(\mathcal{P})$, whence $\pi(\mathcal{P})$ is viewed at full precision. Therefore, it is necessary to fix in advance the way we lift the points of E to p-adics; putting a point of E inside brackets, means that we have lifted the point in this predefined way. Now there are many ways to predefine the lifting of the points of E. However, if we insist on the fact that the map φ is compatible with the group law, there is no choice at all: a point P of E is of a given order n and there is only one point $\langle P \rangle$ that lifts P and remains of order n (unless n is divisible by the characteristic p, but we have excluded that case). Let then define the lift $\langle P \rangle$ of a point P to be the lift that is compatible with the group law. With that definition, the map φ is a group isomorphism and by Theorem 1, φ is efficiently computable.

The reason we are interested in φ is that $\mathcal{E}_1$ can be mapped to the underlying p-adic field and this map is an additive group morphism. Furthermore this morphism is very explicit since it is what is called the p-adic elliptic logarithm that can be computed efficiently using a power expansion in terms of the coordinates of the point, and it converges if the point is in $\mathcal{E}_1$. We denote by φ_1 the map given by the second coordinate of φ, and by log the p-adic elliptic logrithm that sends $\mathcal{E}_1$ to K. We have just explained that $\log \circ \varphi_1$ is an efficiently computable group homomorphism from $\mathcal{E}_0$ to K.

Imagine now that we have two points $\mathcal{P}$ and $\mathcal{Q}$ in $\mathcal{E}$ such that $\mathcal{Q} = \lambda\mathcal{P}$ with an unknown λ. Then we can compute the images of $\mathcal{P}$ and $\mathcal{Q}$ by $\log \circ \varphi_1$ and the homomorphism property ensures that:

$$(\log \circ \varphi_1)(\mathcal{Q}) = \lambda\ (\log \circ \varphi_1)(\mathcal{P}). \tag{1}$$

From this equation, the discrete logarithm λ is easily deduced as the quotient of the two p-adic elliptic logarithms.

The problem in this approach is that in practice we start with points P and Q over the finite field $\mathbb{F}_q$ and in order to apply the above equation, it is necessary to lift them to points $\mathcal{P}$ and $\mathcal{Q}$ defined over p-adics in a way that keeps the relation $\mathcal{Q} = \lambda\mathcal{P}$. The only way I can think of to keep that relation is to use the n-torsion structure as explained in Theorem 1. But then, $\mathcal{P}$ is precisely $\langle P \rangle$ and $\mathcal{Q}$ is $\langle Q \rangle$. Hence $\varphi_1(\mathcal{P}) = \varphi_1(\mathcal{Q}) = 0$, and Equation 1 is just $0 = 0\lambda$, which is obviously useless to compute λ.

Other attempts that I tried in that direction also failed, for instance, modifying φ so that it is a group homomorphism only up to some precision, or to lift the relation $\mathcal{Q} = \lambda\mathcal{P}$ only up to some precision, so that Equation 1 is non-trivial. However, no useful information seems to be obtained.

To conclude, we summarize our discussion in the following theorem:

Theorem 4 *Let K be a p-adic field whose residue class field is the finite field $\mathbb{F}_q$. Let $\mathcal{P}$ and $\mathcal{Q}$ be two points of an elliptic curve $\mathcal{E}$ defined over K, such that there exists an integer λ with $\mathcal{Q} = \lambda\mathcal{P}$. Let n be the order of the reduction of $\mathcal{P}$ in $\mathbb{F}_q$. Then if $\mathcal{P}$ is not an n-torsion point in $\mathcal{E}$, λ can be computed in polynomial time.*

Corollary 1 *To solve a elliptic curve discrete logarithm problem, it is enough to lift it to a p-adic field, in a way that does not preserve the order of the points.*

This corollary looks similar to what we have on $\mathbb{Q}$: if we manage to lift the discrete logarithm problem to a non-torsion subgroup of a curve defined over the rationals, we can deduce the solution using the height function. However, on the rationals, we expect this lifting to be impossible due to the huge height of the points we have to deal with. On the p-adics, the height is no longer a problem.

4 Extensions to other curves

To extend Theorem 1 to higher genus curves, we need to find something to replace the division polynomial ψ_n. In fact, having an algorithm to compute in the Jacobian should be enough. Indeed, assuming that the points are generic, the multiplication by n map is given by a sequence of additions, multiplications, divisions, where all the branching points are fixed. The number of such operations is $O(\log n)$, and furthermore the divisions can be pushed to the end of the process, using analogues to the projective coordinates. Then computing the derivative of the multiplication by m map is a matter of formal manipulations on the sequence of operations. The cost is still in $O(\log n)$ operations. Then the equivalent of Theorem 1 follows, with the same complexity, but a much larger constant hidden in $O()$.

5 Lifting to the complex numbers

The Newton lifting of the n-torsion points should also work in the complex number setting. There are however two problems:

1. Numerical stability;
2. Initialisation.

The problem of the numerical stability is more theoretical than practical: once we are close enough to a root of a polynomial the algorithm should converge pretty quickly. The question of the initialisation is the trickiest and raises the other question: what is the input of the algorithm. If the algorithm is given a point of order n on a curve defined over a finite field, then there is no hope to lift it to $\mathbb{C}$ in a meaningfull way: having a p-adic approximation of the abscissa does not help in having a complex approximation. The only algorithm we can think of is to compute the Deuring lift of the curve to $\overline{\mathbb{Q}}$, then to lift p-adically the point sufficiently to be able to recognize its abscissa as an element of $\overline{\mathbb{Q}}$ (using LLL), and then to compute a floating point approximation of this abscissa. This is of course far slower than whatever direct approach.

Anyway, if by chance reasonnable complex approximations of P and Q were known, then running our Newton lifting based on the n-division polynomial would allow to make these approximation precise enough so that the discrete logarithm could be deduced as the quotient of the abscissae modulo the lattice.

6 Appendix: A trivial remark

We recall here a basic fact which is not specific to elliptic cruve discrete logarithm: if we have an algorithm that can guess (with good probability) one bit of information on the discrete log, then it can be converted in an algorithm that compute the exact discrete logarithm.

Lemma 3 *Let G be a cyclic group generated by P, of known odd order n. Assume that there exists an efficient algorithm $\mathcal{A}$ that takes in input a point Q of G and returns the less significant bit of the discrete logarithm in $[0, n-1]$ of Q with base P. Then, given a point Q in G, it is possible to compute its discrete logarithm with $O(\log n)$ calls to the algorithm $\mathcal{A}$, and a polynomial number of operations in G.*

Proof. Let Q be a point of G and let $\lambda \in [0, n-1]$ be the unknown discrete logarithm. A first call to the algorithm $\mathcal{A}$ returns the less significant bit λ_0 of λ. Then we compute $Q' = (Q - \lambda_0 P)/2$. This division by 2 is possible since the order of P is known and is odd. Then calling algorithm $\mathcal{A}$ with input Q' will return the second less significant bit of λ, and we continue until all the bits are known. □

There are some variants of this Lemma: if the algorithm $\mathcal{A}$ returns the value of λ modulo some integer k different from 2, then the same proof works, as long as n is coprime to k. In practice, we always consider n prime anyway, after the Pollig-Helmann reduction. If the algorithm compute the size of λ (this is more or less the most significant bit, if we always write λ on $\log_2(n)+1$ bits), then the proof can also be trivially adpated. Most of the variants that we can think of (17-th bit, bit in the middle,...) also are enough to reconstruct the whole discrete logarithm.

This Lemma means that if an algorithm based on p-adic lifting can solve the discrete logarithm problem only modulo p, even if p is small it is enough to break completely the discrete logarithm.

References

[1] I. Blake, G. Seroussi, and N. Smart. *Elliptic curves in cryptography*, volume 265 of *London Math. Soc. Lecture Note Ser.* Cambridge University Press, 1999.

[2] H. G. Rück. On the discrete logarithm in the divisor class group of curves. *Math. Comp.*, 68(226):805–806, 1999.

[3] T. Satoh and K. Araki. Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves. *Comment. Math. Helv.*, 47(1):81–92, 1998.

[4] I. A. Semaev. Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curves in characteristic p. *Math. Comp.*, 67(221):353–356, January 1998.

[5] J. H. Silverman. *The arithmetic of elliptic curves*, volume 106 of *Graduate Texts in Mathematics*. Springer–Verlag, 1986.

[6] N. Smart. The discrete logarithm problem on elliptic curves of trace one. *J. of Cryptology*, 12(3):193–196, 1999.