Computing the eigenvalue in the Schoof-Elkies-Atkin algorithm using Abelian lifts

P. Mihăilescu
Mathematisches Institut
der Universität Göttingen
Germany
preda@uni-math.gwdg.de

F. Morain*
LIX, École polytechnique,
Palaiseau, France
morain@lix.polytechnique.fr

É. Schost
ORCCA and Computer
Science Department
University of Western Ontario
London, ON, Canada
schost@scl.csd.uwo.ca

ABSTRACT

The Schoof-Elkies-Atkin algorithm is the best known method for counting the number of points of an elliptic curve defined over a finite field of large characteristic. We use Abelian properties of division polynomials to design a fast theoretical and practical algorithm for finding the eigenvalue.

1. INTRODUCTION

The Schoof-Elkies-Atkin (SEA) algorithm [2] is currently the fastest known algorithm for computing the cardinality of elliptic curves defined over finite fields of large characteristic.

Following the initial work of Schoof [26], considerable work has been devoted to making this algorithm efficient [1, 11, 22, 27, 12, 19, 16]. Elkies's improvement consists in using a factor $f_{\ell,\lambda}$ of degree $(\ell-1)/2$ of the ℓ-th division polynomial, that has degree $(\ell^2-1)/2$. This factor is associated with an eigenspace of the Frobenius; one then searches for the corresponding eigenvalue λ, defined modulo ℓ. The dominant cost of this approach is that of computing $X^p \bmod f_{\ell,\lambda}$, due to the improvements in [19, 16].

In the present article, we describe the complexity aspects and the implementation of a new approach of the first author [21]. We look for $\lambda \in (\mathbb{Z}/\ell\mathbb{Z})^*$ via its logarithm x to some generator $c \bmod \ell$. In turn, we find x modulo coprime divisors q of $\ell-1$. Using Galois theoretic arguments, we find polynomials $M(T)$ and $C(T)$ such that $x \bmod q$ can be obtained by solving $T^p \equiv C^{(v)}(T) \bmod M(T)$ where $C^{(v)}$ is v-fold composition. Since $\deg(M) < \deg(f_{\ell,\lambda})$, the method is faster when q is smaller than $(\ell-1)/2$.

In Section 2, we recall briefly the SEA algorithm. Section 3 describes the use of Abelian lifts for computing the eigenvalue in the Elkies case. In Section 4, we first review classical algorithms that will be used in the implementation and add slight improvements to some variants; we then explain how to implement our algorithms and give examples. Section 5 gives some improvements to our basic scheme and Section 6 concludes the article with timings obtained with our NTL implementation.

*Projet TANC, Pôle Commun de Recherche en Informatique du Plateau de Saclay, CNRS, École polytechnique, INRIA, Université Paris-Sud. The author is on leave from the French Department of Defense, Délégation Générale pour l'Armement.

2. THE SEA ALGORITHM

Throughout the article, E denotes an elliptic curve defined over the finite field $\mathbb{F}_p$ by an equation of the form $Y^2 = X^3 + AX + B$. In all that follows, the X and Y-coordinates of a point P in the affine plane are denoted by subscripts, namely P_X and P_Y. We refer to [2] for the following facts.

2.1 Schoof's original algorithm

There is a group law on an elliptic curve, that is known as the tangent-and-chord method; in particular, the m-fold multiple of a point P is written $[m]P$. Over any field, the addition formulae are rational; repeated use of these rules leads to the introduction of *division polynomials*, generally noted $f_m(X)$. For $m \in \mathbb{N}$, let $E[m]$ denote the m-torsion subgroup of E; then, the division polynomials are used to represent the coordinate ring of $E[m]$ as

$$\mathbb{F}_p[X,Y]/(Y^2 - (X^3 + AX + B), f_m(X)).$$

Let π be the Frobenius endomorphism of $E(\overline{\mathbb{F}_p})$ that sends (X,Y) to (X^p, Y^p). This endomorphism satisfies an equation of the form $\pi^2 - t\pi + p = 0$; the polynomial $T^2 - tT + p$ is called the *characteristic polynomial* of π and the number of $\mathbb{F}_p$-rational points of E is then equal to $p + 1 - t$. Hasse's bound ensures that the absolute value of the *trace* t is bounded by $2\sqrt{p}$. In what follows, we will be interested in *ordinary* curves, for which $t \neq 0$, since the cardinality of supersingular curves is simply $p+1$.

To compute the cardinality of $E(\mathbb{F}_p)$, Schoof's algorithm proceeds by computing t modulo small primes ℓ using the action of π on the set of ℓ-torsion points $E[\ell]$, until enough modular information is known to reconstruct the trace and the cardinality of E by the Chinese Remainder Theorem.

2.2 The Elkies case

In the so-called Elkies case, the characteristic polynomial of π has two linear factors modulo ℓ, so that the restriction of π to $E[\ell]$ has two rational eigenspaces. One of these (call it V) is characterized by a polynomial $f_{\ell,\lambda}(X)$ of degree $(\ell-1)/2$ which divides $f_\ell(X)$; in particular, the coordinate

ring of V can be represented as

$$\mathcal{A} = \mathbb{F}_p[X,Y]/(Y^2 - (X^3 + AX + B), f_{\ell,\lambda}(X)). \quad (1)$$

The action of the Frobenius endomorphism on $(X,Y) \in \mathcal{A}$ is $\pi(X,Y) = (X^p, Y^p)$, with X^p being reduced modulo $f_{\ell,\lambda}(X)$ and the Y^p modulo $f_{\ell,\lambda}(X)$ and $Y^2 - (X^3 + AX + B)$.

We are interested in computing the eigenvalue of π, *i.e.* the integer λ, $0 < \lambda < \ell$ such that $\pi(X,Y) = [\lambda](X,Y)$ or

$$(X^p, Y^p) = [\lambda](X,Y). \quad (2)$$

Indeed, the trace modulo ℓ is then deduced from the formula $t \equiv \lambda + p/\lambda \mod \ell$ (note that $\lambda \neq 0$ for ordinary curves).

We are in the Elkies case if the univariate polynomial $\Phi_\ell(X, j(E))$ has a root in $\mathbb{F}_p$, where $\Phi_\ell \in \mathbb{Z}[X,Y]$ is an equation of the modular curve $X_0(\ell)$ (see [13] for fast algorithms for computing modular polynomials). We will consider the generic case, where the equation $\Phi_\ell(X, j(E)) = 0$ has two roots (when the number of roots is > 2, we know the eigenvalue up to sign). One can then recover the factor $f_{\ell,\lambda}(X)$ of $f_\ell(X)$ in quasi-linear time [3].

Algorithms related to this case may be found in [27, 22]. On a heuristic basis, for a general curve, we expect to be in the Elkies case for about half of the primes ℓ. Combining this with Hasse's bound and the prime number theorem, we obtain that, asymptotically, the largest ℓ we have to consider is about $\log(p)$, where log is the natural logarithm.

We address in this paper the following question: given $f_{\ell,\lambda}(X)$, compute the eigenvalue λ satisfying Equation (2). Several variants are presented in [16] for this task; the fastest algorithm has complexity

$$O(\mathsf{M}(\ell)\log(p) + \sqrt{\ell}\mathsf{M}(\ell)),$$

where $\mathsf{M}(d)$ denotes the cost of multiplying two polynomials of degree less than d in $\mathbb{F}_p[X]$, see Section 4 for details. We present in the next section a different algorithm, which turns out to be faster in many cases.

3. ABELIAN LIFTS

We introduce in this section the main ingredients for our algorithm. Lifting to characteristic zero, we first study the Galois structure of extensions generated by lifts of division polynomials. Reducing back to characteristic p, this enables us to obtain the eigenvalue λ by working in extensions of $\mathbb{F}_p$ of degree q, for q a divisor of $(\ell-1)/2$, as opposed to standard computations which take place modulo $f_{\ell,\lambda}(X)$, that is, in degree $(\ell-1)/2$. Combining the values obtained for coprime divisors of $\ell - 1$ will give us the answer. The theory works for any divisor of $\ell-1$, including even integers. For the even part of the index, one has to consider the Y-coordinates of the torsion points, while the odd part involves X only.

3.1 Lifting to characteristic zero

We start by lifting to characteristic zero; lifts of objects existing modulo p will usually be written with a bar. Let $\mathbb{K}$ be a number field, unramified at ℓ, and let

$$\overline{E} : Y^2 = X^3 + \overline{A}X + \overline{B}$$

be a *non-CM* lift of the curve $E/\mathbb{F}_p$; one can choose in particular $\mathbb{K} = \mathbb{Q}$, the rational field. Then Theorem 4.1 of [28] implies that, for $\ell > 3$, the extension generated by adjoining the coordinates of all ℓ-torsion points of $\overline{E}$ to $\mathbb{K}$ has maximal Galois group, isomorphic to $\mathrm{GL}(2,\ell)$. In particular, the ℓ-division polynomial $\overline{f}_\ell(X)$ associated to $\overline{E}$ is irreducible. We let $\mathbb{K}_\ell = \mathbb{K}[X]/(\overline{f}_\ell(X))$ be the extension of degree $(\ell^2-1)/2$ generated by $\overline{f}_\ell(X)$ and let $\Theta \in \mathbb{K}_\ell$ be the residue class $(X \bmod \overline{f}_\ell(X))$. Let next $\mathbb{L}_\ell$ be the extension

$$\mathbb{L}_\ell = \mathbb{K}_\ell[Y]/\left(Y^2 - (X^3 + \overline{A}X + \overline{B})\right),$$

and let Γ be the image of Y in $\mathbb{L}_\ell$. Consider now the generic point $\overline{P} = (\Theta, \Gamma)$ of $\overline{E}(\mathbb{L}_\ell)$. For $a \in (\mathbb{Z}/\ell\mathbb{Z})^*$, the action

$$\rho_a : \Theta \mapsto \left([a]\overline{P}\right)_X$$

defines an automorphism of $\mathbb{K}_\ell/\mathbb{K}$ and

$$G = \{\rho_a : 1 \leq a \leq (\ell-1)/2\}$$

is a cyclic subgroup of the Galois group $\mathrm{Gal}(\mathbb{K}_\ell/\mathbb{K})$. Let $\mathbb{K}_0$ be the fixed field $\mathbb{K}_\ell^G$. Then, the extension $\mathbb{K}_\ell/\mathbb{K}_0$ is cyclic of degree $(\ell-1)/2$ and the polynomial

$$\overline{f}_{\ell,\lambda}(T) = \prod_{a=1}^{(\ell-1)/2} (T - \rho_a(\Theta)) \in \mathbb{K}_0[T]$$

is a cyclic polynomial for which $\mathbb{K}_\ell = \mathbb{K}_0[T]/\left(\overline{f}_{\ell,\lambda}(T)\right)$. For a in $(\mathbb{Z}/\ell\mathbb{Z})^*$, define the unique polynomial $\overline{g}_a \in \mathbb{K}_0[X]$ by the condition that $\deg(\overline{g}_a(X)) < (\ell-1)/2$ and $\overline{g}_a(\Theta) = \rho_a(\Theta)$. We have the composition rules

$$\overline{g}_a(\overline{g}_b(\Theta)) = \overline{g}_b(\overline{g}_a(\Theta)) = \overline{g}_{ab}(\Theta) = \rho_{ab}(\Theta)$$

and in particular, we can rewrite

$$\overline{f}_{\ell,\lambda}(T) = \prod_{a=1}^{(\ell-1)/2} (T - \overline{g}_a(\Theta)) \in \mathbb{K}_0[T].$$

Note that the set $\{\rho(\Theta) : \rho \in G\}$ is a normal basis of $\mathbb{K}_\ell/\mathbb{K}_0$. In particular, for each subfield $\mathbb{K}_0 \subset \mathbb{K}' \subset \mathbb{K}_\ell$, the trace $\mathbf{Tr}_{\mathbb{K}_\ell/\mathbb{K}'}(\Theta)$ together with its conjugates form a normal basis for $\mathbb{K}'/\mathbb{K}_0$ and generates the field $\mathbb{K}'$ as a simple extension.

The basic idea of our approach is to consider traces in several such subextensions $\mathbb{K}_0 \subset \mathbb{K}' \subset \mathbb{K}_\ell$ of coprime degrees $q = [\mathbb{K}' : \mathbb{K}_0]$. We will thus write $\overline{\tau}_q = \mathbf{Tr}_{\mathbb{K}_\ell/\mathbb{K}'}(\Theta)$, so that $\mathbb{K}' = \mathbb{K}_0[\overline{\tau}_q]$ by what was said previously. For a given q, the action of the Frobenius on these $\overline{\tau}_q$ will allow us to compute the residue modulo q of the index of λ in $(\mathbb{Z}/\ell\mathbb{Z})^*$; using Chinese Remaindering will yield λ.

For q odd, the previous setting is enough, but to treat the case q even, we have to extend the discussion and consider Y-coordinates. The extension $\mathbb{L}_\ell = \mathbb{K}_\ell[\Gamma]$ has degree 2, with Galois group generated by $\varsigma : \Gamma \mapsto -\Gamma$. The automorphism ς acts naturally on $\mathbb{K}_\ell$ as the identity, making it an element $\varsigma \in \mathrm{Gal}(\mathbb{L}_\ell/\mathbb{K}_0)$. Let $F(X) = X^3 + \overline{A}X + \overline{B}$ and define

$$S(Y) = \prod_{a=1}^{(\ell-1)/2} (Y^2 - F(\rho_a(\Theta)) \in \mathbb{K}_0[Y].$$

Algebraic verifications show that $\mathbb{L}_\ell = \mathbb{K}_0[Y]/(S(Y))$ and that $S(Y)$ splits completely over $\mathbb{K}_0$, making the extension $\mathbb{L}_\ell/\mathbb{K}_0$ Galois. From the theory of division functions, one gathers that there are polynomials $h_b(X) \in \mathbb{K}_0[X]$ such that

$$\left([b]\overline{Q}\right)_Y = \overline{Q}_Y \cdot h_b\left(\overline{Q}_X\right), \quad \forall\, \overline{Q} \in \langle\overline{P}\rangle.$$

Note that $h_b(X) = -h_{\ell-b}(X)$ as a consequence of the parities of the X,Y-coordinates. We claim that $\mathbb{L}_\ell/\mathbb{K}_0$ is Abelian and in fact that

$$H = \mathrm{Gal}(\mathbb{L}_\ell/\mathbb{K}_0) = \langle\varsigma\rangle \times G.$$

For this, we show how to lift ρ_a to $\tilde{\rho}_a \in H$; we do this for $a = c$, a generator of $(\mathbb{Z}/\ell\mathbb{Z})^*$, by the natural definition:

$$\tilde{\rho}_c(\Gamma) = \left([c]\overline{P}\right)_Y = \Gamma \cdot h_c(\Theta).$$

Note first that $\tilde{\rho}_c(\Gamma^2) = F(g_c(\Theta)) = F(\Theta)\cdot h_c(\Theta)^2$, which shows that the restriction of $\tilde{\rho}_c$ on $\mathbb{K}_\ell$ acts indeed like ρ_c. One then verifies that $\varsigma\circ\tilde{\rho}_c = \tilde{\rho}_c\circ\varsigma$ and that it generates an automorphism group acting on $\mathbb{L}_\ell/\mathbb{L}_0$, where $[\mathbb{L}_0:\mathbb{K}_0]=2$ (we refer to [21] for the details). Now, for an intermediate field $\mathbb{K}_0\subset\mathbb{K}'\subset\mathbb{K}_\ell$ as above, with $q=[\mathbb{K}':\mathbb{K}_0]$, observe the existence of an intermediate *ordinate*-field

$$\mathbb{L}' = \mathbb{K}'[\overline{\tau'}_q] \quad \text{with} \quad \overline{\tau'}_q = \textstyle\sum_{a=1}^{(\ell-1)/2q} \tilde{\rho}_{c^{aq}}(\Gamma),$$

which is an extension of degree 2 of $\mathbb{K}'$, with $\overline{\tau'}_q{}^2 \in \mathbb{K}'$ (the field $\mathbb{L}_0$ above is one of these fields).

We finally introduce elliptic Gaussian periods. As before, we assume that c generates $(\mathbb{Z}/\ell\mathbb{Z})^*$. We suppose first that q is an odd divisor of $(\ell-1)$ and develop this case in more detail; in this case, we need only work with X-coordinates. We then let $q'=(\ell-1)/(2q)$, $h=c^q$, $k=c^{q'}$ and

$$(\mathbb{Z}/\ell\mathbb{Z})^*/\{\pm1\} = H\times K \quad \text{with} \quad H=\langle h\rangle,\quad K=\langle k\rangle.$$

For $0\le i<q$, we define

$$\overline{\eta}_i \;=\; \textstyle\sum_{a\in H}\left([k^i\cdot a]\overline{P}\right)_X = \sum_{a\in H}\rho_a\left(\rho_{k^i}(\Theta)\right);$$

then, $\overline{\eta}_0$ is the trace $\overline{\tau}_q$ defined above, and $\overline{\eta}_i = \rho_k^{(i)}(\overline{\eta}_0)$ for all i. As a consequence, there is a cyclic action:

$$\overline{\eta}_0 \xrightarrow{\rho_k} \overline{\eta}_1 \xrightarrow{\rho_k} \cdots \xrightarrow{\rho_k} \overline{\eta}_{q-1} \xrightarrow{\rho_k} \overline{\eta}_0, \tag{3}$$

so that the minimal polynomial of $\overline{\eta}_0$ is

$$\overline{M}(T) = \prod_{i=0}^{q-1}(T-\overline{\eta}_i) \in \mathcal{O}(\mathbb{K}_0)[T]$$

and $\mathbb{K}' = \mathbb{K}_0[T]/(\overline{M}(T))$. Since the extension $\mathbb{K}'/\mathbb{K}_0$ is cyclic, there is a polynomial $\overline{C}\in\mathbb{K}_0[T]$ which encodes the action of ρ_k, so $\overline{C}(\overline{\eta}_0)=\rho_k(\overline{\eta}_0)=\overline{\eta}_1$. Iterating the construction gives $\overline{\eta}_i = \overline{C}^{(i)}(\overline{\eta}_0)$, where the exponent (i) denotes i-fold composition of the polynomial C with itself.

In the case when q is even, we let $q'=(\ell-1)/q$ and h,k,H,K like previously, noting that this time $H\times K = (\mathbb{Z}/\ell\mathbb{Z})^*$. We work accordingly in $\mathbb{L}'$ instead of $\mathbb{K}'$ and define

$$\overline{\eta'}_i = \textstyle\sum_{a\in H}\left([k^i\cdot a]\overline{P}\right)_Y = \sum_{a\in H}\tilde{\rho}_a\left(\tilde{\rho}_{k^i}(\Gamma)\right);$$

then, $\overline{\eta'}_0$ is the trace $\overline{\tau'}_q$ defined above, and $\overline{\eta'}_i = \tilde{\rho}_k^{(i)}(\overline{\eta'}_0)$ for all i. Thus, there is now a two-phased cyclic action:

$$\overline{\eta'}_0 \xrightarrow{\tilde{\rho}_k} \overline{\eta'}_1 \xrightarrow{\tilde{\rho}_k} \cdots \xrightarrow{\tilde{\rho}_k} \overline{\eta'}_{q/2-1} \xrightarrow{\varsigma} \overline{\eta}_{q/2} \xrightarrow{\tilde{\rho}_k} \cdots \xrightarrow{\tilde{\rho}_k} \overline{\eta'}_{q-1} \xrightarrow{\tilde{\rho}_k} \overline{\eta'}_0. \tag{4}$$

Since $\eta_{q/2+j}=-\eta_j$, the minimal polynomial of $\overline{\eta'}_0$ is

$$\overline{M}(T) = \prod_{i=0}^{q/2-1}\left(T^2-(\overline{\eta'}_i)^2\right) = \overline{N}(T^2)\in\mathcal{O}(\mathbb{K}_0)[T].$$

and $\mathbb{L}'=\mathbb{K}_0[T]/(\overline{M}(T))$. The extension $\mathbb{L}'/\mathbb{K}_0$ is Abelian, yet not cyclic. Note that $\overline{\eta'}_j/\overline{\eta'_0}\in\mathbb{K}'$, since one checks it is invariant under ς. But $(\overline{\eta'}_0)^2$ generates $\mathbb{K}'/\mathbb{K}_0$ and thus

$$\tilde{\rho}(\overline{\eta}_0) = \overline{\eta'}_0\cdot\overline{D}(\overline{\eta'}_0^2),$$

for some polynomial $\overline{D}\in\mathbb{K}_0[X]$. The action of $\tilde{\rho}_k$ is thus encoded in this case by a polynomial $\overline{C}(T)=T\cdot\overline{D}_1(T)\in\mathbb{K}_0[T]$, with $\overline{D}=D_1^{(k)}$, so that we have

$$\overline{C}(\overline{\eta'}_0)=\tilde{\rho}_k(\overline{\eta'}_0)=\overline{\eta'}_1 \quad\text{and}\quad \overline{\eta'}_{q-1-i}=\varsigma\left(\overline{\eta'}_i\right)=-\overline{\eta'}_i.$$

Let us define

$$\begin{array}{llll} j(i)=i & \text{and} & s(i)=1 & \text{for} \quad i<q/2 \\ j(i)=q-1-i & \text{and} & s(i)=-1 & \text{for} \quad q/2\le i<q. \end{array}$$

With this we can group the previous to $\overline{\eta'}_i = s(i)\cdot\overline{C}^{(j(i))}(\overline{\eta'}_0)$.

3.2 Finding the eigenvalue

We can then reduce the previous construction back to $\mathbb{F}_p$. Let us consider the algebras $\mathcal{A}_0=\mathbb{F}_p[X]/(f_{\ell,\lambda}(X))$ and $\mathcal{A}$ (defined in Equation (1)). Let θ and γ be the residue classes of X and Y in $\mathcal{A}$ and let $P=(\theta,\gamma)$ be a "generic" point on E. For $a\in(\mathbb{Z}/\ell\mathbb{Z})^*$, we define the unique *multiplication by* a-polynomial of $\mathcal{A}$ by the condition $\deg(g_a(X))<(\ell-1)/2$ and $g_a(\theta)=([a]P)_X\in\mathcal{A}$. Remark that the eigenvalue λ satisfies the relation $\theta^p=g_\lambda(\theta)$.

Since ℓ is an Elkies prime, there is a prime ideal $\mathfrak{p}\subset p\cdot\mathcal{O}(\mathbb{K}_0)$ such that $\overline{f}_{\ell,\lambda}(X) \bmod \mathfrak{p} = f_{\ell,\lambda}(X)$; the polynomial $f_{\ell,\lambda}(X)$ has thus $\overline{f}_{\ell,\lambda}(X)\in\mathcal{O}(\mathbb{K}_0)[X]$ as a cyclic lift. Similarly, for all a, $\overline{g}_a(X)$ is a lift of $g_a(X)$. The definition of the multiplication polynomials $g_a(X)$ then shows that we have

$$f_{\ell,\lambda}(Z) = \textstyle\prod_{a=1}^{(\ell-1)/2}(Z-g_a(\theta)).$$

We have depicted the situation in Figure 1.

Let as above c be a generator of $(\mathbb{Z}/\ell\mathbb{Z})^*$. We denote by x the index of λ, so that $\lambda=c^x \bmod \ell$. Using the construction of the previous subsection, we describe now how to recover $(x \bmod q)$, giving details for q an odd divisor of $(\ell-1)/2$. We refer to Subsection 4.2 for a description of the algorithm in the case q even. For $0\le i<q$, define

$$\eta_i = \textstyle\sum_{a\in H} g_a\left(g_{k^i}(\theta)\right),$$

so that in particular $\eta_i=\overline{\eta}_i \bmod \mathfrak{p}$. For odd $\ell\ge3$, the discriminant of f_ℓ satisfies the following relation (see [9]):

$$\mathrm{Disc}(f_\ell) = (-1)^{(\ell-1)/2}\ell^{(\ell^2-3)/2}(-\Delta)^{(\ell^2-1)(\ell^2-3)/24},$$

where $\Delta(E)=-2^4(4A^3+27B^2)$. We can thus safely assume that all roots of f_ℓ, and thus of $f_{\ell,\lambda}$, are distinct. However, the traces $\overline{\eta}\in\mathbb{K}'$ do not generate a normal base and thus the discriminant $\Delta'=\mathrm{Disc}(\overline{\eta})$ may be divisible by some primes $(p',\mathrm{Disc}(f_\ell))$. We make the additional assumption that the minimal polynomial $\overline{M}(X)$ of $\overline{\eta}_0$ remains separated upon reduction modulo $\mathfrak{p}$.

We let $M(X)\in\mathbb{F}_p[X]$ be the minimal generating polynomial of the sequence of powers of $\eta_0\in\mathcal{A}_0$ and note that by the above condition on p, we have $M(X)=\overline{M}(X) \bmod \mathfrak{p}$; hence, $M(X)$ has degree q.

Writing $C(T)=\overline{C}(T) \bmod \mathfrak{p}$, we also have the relation $\eta_i=C^{(i)}(\eta_0)$, by reduction modulo $\mathfrak{p}$ of the relation holding in $\mathbb{K}'$. In the even case, one defines $\eta'_i=\overline{\eta'_i} \bmod \mathfrak{p}$, reduces $C'(T)=T\cdot\overline{D}(T) \bmod \mathfrak{p}$ and has accordingly $\eta'_i/\eta'_0 = D^{(i)}(T^2)$. Note that $N(T^2)=0$ and thus one may use this polynomial in relation with the action of $\tilde{\rho}$.

The importance of the Abelian lift consists in granting the existence of the polynomials $C(T)$ which are not describing automorphisms in a field theoretic sense, but are images of automorphisms from the lift. In a general theory of Galois extensions of rings, see e.g. [18, 20], we encounter $C(X)$ as automorphisms of rings (algebras).

Let us finally show how to recover $(x \bmod q)$. There exists $v\in\mathbb{Z}/q\mathbb{Z}$ with $T^p=C^{(v)}(T) \bmod M(T)$, so that

$$\eta_0^p = C^{(v)}(\eta_0) = \eta_v.$$

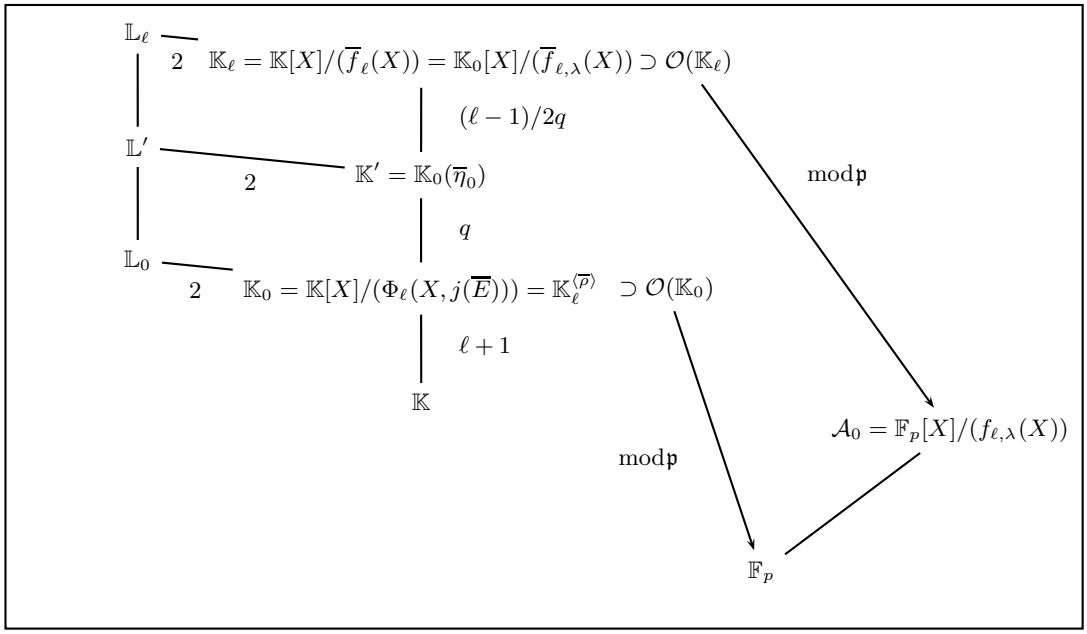

Figure 1: Extension fields.

But $\eta_0^p = \eta_\lambda$, extending the index of η modulo q. Thus, $c^x = c^{q'v} \bmod \ell$ or $x = q'v \bmod q$. In the even case, one can work with polynomials of degree $q/2$, as shown above.

4. ALGORITHMS AND COMPLEXITY

We give here the details of the algorithm and its complexity analysis. The notation $\mathsf{M}(n)$ denotes the time needed to multiply polynomials of degrees less than n [14, Chapter 8]. We make the super-linearity assumption that $\mathsf{M}(n+n') \geq \mathsf{M}(n) + \mathsf{M}(n')$ holds for all n, n'. Over fields supporting FFT, one can take $\mathsf{M}(n) \in O(n\log(n))$; in all cases, one has $\mathsf{M}(n) \in O(n\log(n)\log\log(n))$, by the results of [25, 5]. Our algorithms aslo rely on matrix operations. We will denote by $\omega < 3$ an exponent such that matrices in $\mathbb{F}_p^{n\times n}$ can be multiplied in $O(n^\omega)$ operations. The current record is Coppersmith and Winograd's $\simeq 2.38$ exponent [6].

The *minimal polynomial* of an element α in a finite-dimensional $\mathbb{F}_p$-algebra $\mathcal{A}$ is the minimal generating polynomial of the sequence $(\alpha^i)_{i\geq 0}$ (it is not necessarily irreducible).

4.1 Preliminaries

Most results given here are known; those not in the literature are straightforward generalizations of existing ones.

Modular composition and related problems. Let $\mathcal{C}(n)$ be an upper bound on the cost of computing $g(h) \bmod f$, where f, g, h are in $\mathbb{F}_p[X]$, of degrees n. Using Brent-Kung's modular composition algorithm [4], one can take

$$\mathcal{C}(n) \in O(n^{1/2}\mathsf{M}(n) + n^{(\omega+1)/2});$$

we will assume that $\mathsf{M}(n)\log(n) \in O(\mathcal{C}(n))$. When g has degree $q \leq n$, the bound reduces to

$$O(q^{1/2}\mathsf{M}(n) + q^{(\omega-1)/2}n).$$

We need two variants of this algorithm. We write $\mathcal{C}_r(n)$ for the cost of performing r modular compositions $\{g_i(h) \bmod f\}_{1\leq i\leq r}$, where f and h are fixed. For $r \in O(n)$, using the algorithm of [30, 17], one can take

$$\mathcal{C}_r(n) \in O(r^{1/2}n^{1/2}\mathsf{M}(n) + r^{(\omega-1)/2}n^{(\omega+1)/2}).$$

The other variant is iterated composition: given $k \in O(n)$, compute $h, h(h) \bmod f, \ldots, h^{(k)} \bmod f$, assuming that f divides $f(h)$. We use the algorithm of [17, Lemma 4]: if

$$H_1 = h, H_2 = h(h) \bmod f, \ldots, H_j = h^{(j)} \bmod f$$

are known, we deduce

$$H_{j+1} = H_1(H_j) \bmod f, \ldots, H_{2j} = H_j(H_j) \bmod f.$$

We repeat this scheme for $j = 1, \ldots, 2^{\lceil\log(k)\rceil}$. The total cost is up a constant that of the last step, *i.e.*, $O(\mathcal{C}_k(n))$.

Minimal polynomials and related problems. Let $f \in \mathbb{F}_p[X]$ be of degree n and let α be in $\mathcal{A} = \mathbb{F}_p[X]/(f)$. Given a linear form $\mathcal{L} : \mathcal{A} \to \mathbb{F}_p$ and $q \leq n$, the sequence $[\mathcal{L}(\alpha^i)]_{0\leq i\leq q}$ can be computed in time $\mathcal{C}(n)$ see [29, 31]; a more precise bound is

$$O(q^{1/2}\mathsf{M}(n) + q^{(\omega-1)/2}n).$$

Thus, if the minimal polynomial of α has degree q, it can be computed within the same complexity [29, 31].

Let now β be in $\mathcal{A}$, and suppose that there exists $C \in \mathbb{F}_p[X]$ of degree less than q such that $\beta = C(\alpha)$, where q is the degree of the minimal polynomial of α. In [29, Theorem 5], Shoup gave an algorithm of complexity $O(\mathcal{C}(n))$ for computing C, under the condition that f is irreducible.

This algorithm could be extended to the general case by using randomization. We present a different solution using the trace form as in [24, 23], which applies in characteristic $p > n$; we also mention how to obtain an upper bound of

$$O(q^{1/2}\mathsf{M}(n) + q^{(\omega-1)/2}n).$$

For simplicity, we assume that the characteristic polynomial $\chi(X)$ of α is a power of its minimal polynomial $M(X)$, say

$\chi(X) = M(X)^r$. We also assume that $f(X)$ is squarefree (all these assumptions are satisfied below).

Let $\theta \in \mathcal{A} = (X \bmod f)$ and let $\mathbf{Tr}$ be the trace $\mathcal{A} \to \mathbb{F}_p$. The values $\mathbf{Tr}(\theta^i)$ for $0 \le i < n$ can be computed in $O(\mathsf{M}(n))$ operations, by expanding f'/f at infinity. Using the "transposed multiplication" algorithm of [31], one can then compute in $O(\mathsf{M}(n))$ operations the linear form $\mathcal{L} : \mathcal{A} \to \mathbb{F}_p$ such that $\mathcal{L}(u) = \mathbf{Tr}(\beta u)$ holds for all $u \in \mathcal{A}$. Then following [24, 23], one sees that

$$\sum_{i=0} \frac{\mathcal{L}(\alpha^i)}{X^{i+1}} = r\frac{C(X)}{M(X)}$$

holds. The sequence $(\mathcal{L}(\alpha^i))_{i \le q}$ can be computed in time $O(q^{1/2}\mathsf{M}(n) + q^{(\omega-1)/2}n)$. Then, C can be recovered in $\mathsf{M}(n)$ operations, proving our claim.

4.2 Main algorithm

We return to the context of the previous sections. Let V be an eigenspace of $E[\ell]$ associated to an eigenfactor $f_{\ell,\lambda}(X)$. We now describe the details of our algorithm, starting in the case q odd and indicating the modifications for q even.

Compositions. Write as before $\mathcal{A}_0 = \mathbb{F}_p[X]/(f_{\ell,\lambda})$, and let θ be the residue class of X in $\mathcal{A}_0$. Given $R = R_X(\theta)$ and $S = S_X(\theta)$ in $\mathcal{A}_0$, we write $R[S] = T_X(\theta)$, with $T_X = R_X(S_X) \bmod f_{\ell,\lambda}$. In particular, if $R_X = g_a$ and $S_X = g_b$ (as defined in Subsection 3.2), then $T_X = g_{ab}$. The cost of computing $R[S]$ is thus of $\mathcal{C}(\ell)$ base field operations.

Computing η_0. We continue with the previous notations. As in Section 3, we let c be a generator of $(\mathbb{Z}/\ell\mathbb{Z})^*$, let q be an odd divisor of $(\ell-1)/2$ and define

$$q' = (\ell-1)/(2q), \quad h = c^q \bmod \ell, \quad k = c^{q'} \bmod \ell.$$

For b in $\mathbb{N}$, define

$$T_{0,X}(h,b) = \textstyle\sum_{j=0}^{b-1} g_{h^j}(\theta),$$

the subscript X showing that we consider only X-coordinates. We compute $\eta_0 = T_{0,X}(h,q')$ by adapting Algorithm 5.2 in [15]. Using the relation

$$T_{0,X}(h, b+b') = T_{0,X}(h,b)[g_{h^{b'}}] + T_{0,X}(h,b'), \qquad (5)$$

we are led to the following divide-and-conquer algorithm, where, at step i, we have $U = g_{h^{2^i}}$, $V = T_{0,X}(h,2^i)$ and $W = T_{0,X}(h, b \bmod 2^i)$.

```
T_{0,X}(h,b)
1    U ← g_h;
2    V ← θ;
3    W ← 0;
4    while b > 0 do
4.1      if b is even then
4.1.1        (U,V,W) ← (U[U], V[U]+V, W);
4.2      else
4.2.1        (U,V,W) ← (U[U], V[U]+V, W[U]+V);
4.3      b ← b/2;
5    return W;
```

The total cost is in

$$O(\mathsf{M}(\ell)\log(\ell) + \mathcal{C}(\ell)\log(q')) \subset O(\mathcal{C}(\ell)\log(q'));$$

in the left-hand estimate, the first term gives the cost of computing g_h in Step 1 by repeated doubling and the second term accounts for the loop in Step 5.

Computing η_1. We continue with the computation of $\eta_1 = \eta_0[g_k]$. We denote by $T_{1,X}(\eta_0)$ a function that performs this operation; its cost is in

$$O(\mathsf{M}(\ell)\log(\ell) + \mathcal{C}(\ell)) \subset O(\mathcal{C}(\ell)),$$

where in the left-hand estimate, the first term accounts for the cost of computing g_k by repeating doubling.

Finding $x \bmod q$. Knowing $T_p = T^p \bmod M$, we will find v such that $T_p = C^{(v)} \bmod M$ using baby steps / giant steps. Writing $v = i + jr$ with $r = \lfloor\sqrt{q}\rfloor$, we look for $0 \le i < r$, $0 \le j \le r$ such that $T_p = C^{(i+jr)} \bmod M$. Let $\tilde{C} = C^{(q-r)} \bmod M$, so that $C^{(r)} \circ \tilde{C} = T \bmod M$. Rewriting the previous equality as $\tilde{C}^{(j)} \circ T_p = C^{(i)} \bmod M$ leads to the following algorithm, of complexity $O(\mathcal{C}_{\sqrt{q}}(q))$.

```
1    C_i ← C^(i) mod M, 0 ≤ i ≤ r;
2    C̃ ← C^(q−r) mod M;
3    C̃_j ← C̃^(j) mod M, 0 ≤ j ≤ r;
4    T_{p,j} ← T_p ∘ C̃_j mod M, 0 ≤ j ≤ r;
5    Find i,j such that T_{p,j} = C_i and return i + jr.
```

Main algorithm. We can now give the details of our main algorithm. Letting x be the index of λ in $(\mathbb{Z}/\ell\mathbb{Z})^*$, so that $\lambda = c^x$, this algorithm computes $x \bmod q$.

```
LogEigenvalue(q, ℓ, f_{ℓ,λ})
1    q' ← (ℓ−1)/(2q);
2    η_0 ← T_{0,X}(h, q');
3    η_1 ← T_{1,X}(η_0);
4    M ← MinimalPolynomial(η_0) ∈ F_p[T];
5    compute C such that η_1 = C(η_0).
6    T_p ← T^p mod M.
7    find 0 ≤ v < q such that T_p = C^(v) mod M(T).
8    return q'v mod q.
```

PROPOSITION 4.1. *The previous algorithm has complexity*

$$O(\mathcal{C}(\ell)\log(q') + \mathsf{M}(q)\log(p) + \mathcal{C}_{\sqrt{q}}(q)).$$

PROOF. It follows from the results given previously. □

There are two extreme cases to consider. If $q \ll \ell$, the dominant step is Step 2, of cost $O(\mathcal{C}(\ell)\log(\ell))$. When $q \approx \ell$, the dominant term is that of Steps 6 and 7; this is no better than standard methods, which rely on computing Y^p and X^p modulo $f_{\ell,\lambda}(X)$ and $Y^2 - (X^3 + AX + B)$.

Modifications for q even. For the case $q = 2$, we know the value of $x \bmod 2$ (the sign of λ, actually) using Dewaghe's trick [10]. For q even > 2, we work in the algebra $\mathcal{A}$ of Equation (1), extending the previous constructions to pairs of polynomials. Let θ and γ be the residue classes of X and Y in $\mathcal{A}$. Given $R = (R_X(\theta), \gamma R_Y(\theta))$ and $S = (S_X(\theta), \gamma S_Y(\theta))$ in $\mathcal{A}^2$, we define $R[S] = (T_X(\theta), \gamma T_Y(\theta))$, with

$$T_X = R_X(S_X) \bmod f_{\ell,\lambda}, \quad T_Y = S_Y \cdot R_Y(S_X) \bmod f_{\ell,\lambda}.$$

Computing $R[S]$ is thus slightly more expensive than in the case q odd, the cost being $\mathcal{C}_2(\ell) + \mathsf{M}(\ell)$. Remark that, writing $P = (\theta, \gamma)$, if $R = [a]P$ and $S = [b]P$, then $R[S] = [ab]P$.

Next, we extend the definition of the traces η_0 and η_1 to take into account ordinates. We are led to compute

$$\eta'_i \;=\; \textstyle\sum_{j=0}^{q'-1} \left([h^j k^i]P\right)_Y,$$

where q' is now defined as $(\ell-1)/q$. We thus define the vector analogue of the previous function $T_{0,X}$, namely

$$T_0(h,b) = \left(\sum_{j=0}^{b-1}([h^j]P)_X, \sum_{j=0}^{b-1}([h^j]P)_Y\right).$$

In this case, Equation (5) becomes

$$T_0(h,b+b') = T_0(h,b)[[h^{b'}]P] + T_0(h,b'),$$

addition being done componentwise. The algorithm to compute $\eta_0' = T_0(h,q')$ is the similar, with initial values $U = [h]P$, $V = P$ and $W = (0,0)$. The computation of η_1' is similar, using vectorial composition of $T_0(h,q')$ by $[k]P$.

In the main algorithm, we then look for the minimal polynomial N of degree $q/2$ of $(\theta^3 + A\theta + B){\eta_0'}^2$, which gives $M(T) = N(T^2)$. The polynomial C of Step 5 now has the form $C(T) = TD(T^2)$. We compute D first; this is done by using the relation $C(\eta_0') = \eta_1' \bmod f_{\ell,\lambda}$ gives

$$D((\theta^3 + A\theta + B){\eta_0'}^2) = \eta_1'/\eta_0' \bmod f_{\ell,\lambda}(X).$$

To perform Step 6, we remark that if $A(U) = U^{(p-1)/2} \bmod N(U)$ holds, then $TA(T^2) = T^p \bmod M(T)$.

General view of the complexity. Summing the contributions of all q dividing $\ell-1$, we see that the complexity of our approach has two components: the contribution of the trace computations will be bounded by the sum of terms of the form $O(\mathcal{C}(\ell)\log(\ell))$; the second component is the sum of terms of the form $O(\mathsf{M}(q)\log(p) + \mathcal{C}_{\sqrt{q}}(q))$, corresponding to the last part of the algorithm. If the largest prime power q dividing $\ell-1$ is small, we expect to be faster than standard approaches, whose cost is dominated by the $O(\mathsf{M}(\ell)\log(p))$ term for computing Y^p and X^p in $\mathcal{A}$.

4.3 Numerical examples

Take $E : Y^2 = X^3 + X + 22$ over $\mathbb{F}_p$, with $p = 1009$. The prime $\ell = 13$ is of Elkies type, having the eigenfactor

$$f_{13,\lambda} = X^6 + 613X^5 + 898X^4 + 703X^3 + 487X^2 + 35X + 770.$$

Let $\theta = X \bmod f_{13,\lambda}$. Since $\ell - 1 = 2^2 \cdot 3$ and $(\mathbb{Z}/13\mathbb{Z})^*$ is generated by $c = 2$, we find, for $q = 3$ and $q' = 2$

$$\begin{aligned}
\eta_0 &= 488\theta^5 + 532\theta^4 + 926\theta^3 + 618\theta^2 + 610\theta + 518,\\
M(T) &= T^3 + 613T^2 + 128T + 550,\\
\eta_1 &= 392\theta^5 + 561\theta^4 + 685\theta^3 + 125\theta^2 + 18\theta + 294,\\
C(T) &= 524T^2 + 394T + 24,\\
T_p &= 524T^2 + 394T + 24,
\end{aligned}$$

so that $v = 1$ and $x = 1 \cdot 2 \bmod 3 = 2 \bmod 3$. For $q = 4$, we have $q' = 3$, so that $h = c^q = 3$, $k = c^{q'} = 8$. We compute

$$\eta_0' = 56\theta^5 + 669\theta^4 + 545\theta^3 + 185\theta^2 + 62\theta + 860$$

from which we get

$$(\theta^3+\theta+22){\eta_0'}^2 = 834\theta^5 + 167\theta^4 + 203\theta^3 + 121\theta^2 + 727\theta + 567$$

whose minimal polynomial is $N(T) = T^2 + 898T + 587$, giving $M(T) = T^4 + 898T^2 + 587$. Now:

$$\eta_1' = 118\theta^5 + 972\theta^4 + 554\theta^3 + 725\theta^2 + 359\theta + 986.$$

We deduce $D(T) = 767T + 241$ and $C(T) = 767T^3 + 241T$. We compute $T^p = 767T^3 + 241T \bmod M(T)$, so $v = 1$ and $x = 3 \cdot 1 \bmod 4$. Combining all information leads to $\lambda = 7$.

5. IMPROVEMENTS

5.1 Combining values of q

The algorithm does not require the values of q to be prime powers: we just need a decomposition of $\ell - 1$ as a product of pairwise coprime numbers. Using larger values of q's tend to diminish the cost of the fast trace algorithm, but we have to balance with the cost of the latter steps.

It is difficult to anticipate what factorization of $\ell - 1$ into coprimes should be used. Write $\ell - 1 = 2^r q_1 \cdots q_s$ with $r \geq 1$ and $q_1 < q_2 < \cdots < q_s$ coprime prime powers (for $\ell < 11,000$, the domain of feasibility of SEA as of now, we have $r \leq 9$ and $s \leq 4$). For $\ell - 1$ of the form 2^r, $2^r q_1$ or $2q_1q_2$, we have no flexibility. In the case $\ell - 1 = 2q_1 \cdots q_s$, we rewrite this as $2q_1'q_2'$ with q_1' as close as q_2' as possible. For $\ell - 1 = 2^r q_1 \cdots q_s$ with $r > 1$, the situation is more involved, since the cost for evaluating η_0 for even q is roughly twice that for odd q. When $\ell - 1 = 2^r q_1 q_2$ with $q_1 < q_2$, if $q_2 \ll \log(p)$, then the dominant cost will be that of modular compositions to compute the various η_0's. The cost of computing all η_0's independently will then approximately be $L_1\mathcal{C}(\ell)$, with

$$\begin{aligned}
L_1 &= 2\log\left((\ell-1)/2^r\right) + \log\left((\ell-1)/q_1\right)\\
&\qquad + \log\left((\ell-1)/q_2\right)\\
&= 4\log(\ell-1) - \log(2^{2r}q_1q_2).
\end{aligned}$$

With $q_1' = 2^r q_1, q_2' = q_2$, the cost will be $\simeq L_2\mathcal{C}(\ell)$, with

$$L_2 = 4\log(\ell-1) - \log(2^{2r}q_1^2q_2),$$

which is smaller. On the other hand, the cost of Step 6 increases from $(\mathsf{M}(2^r) + \mathsf{M}(q_1) + \mathsf{M}(q_2))\log(p)$ to $(\mathsf{M}(2^r q_1) + \mathsf{M}(q_2))\log(p)$. See Section 6 for examples.

Remark further that for a given choice of q_1, q_2, savings are possible. Suppose for instance we are in the situation $\ell - 1 = q_1q_2$, where q_1 is even and q_2 odd; in particular, $q_1' = q_2$ and $q_2' = q_1/2$. For q_1, we have to compute

$$\begin{aligned}
&{\eta_0'}^{(q_1)} = \textstyle\sum([(c^{q_1})^i]P)_Y\\
\text{and}\quad &{\eta_1'}^{(q_1)} = ([c^{q_2}]P)_Y \cdot \left({\eta_0'}^{(q_1)} \circ ([c^{q_2}]P)_X\right) \mod f_{\ell,\lambda}.
\end{aligned}$$

In a symmetric way, for q_2, we need $([c^{q_2}]P)_X$ to compute ${\eta_0}^{(q_2)}$, followed by

$${\eta_1}^{(q_2)} = \eta_0^{(q_2)} \circ ([c^{q_1/2}]P)_X \mod f_{\ell,\lambda}.$$

Therefore, we can amortize the costs of $[c^{q_1}]P$ and $[c^{q_2}]P$, by computing $[c^{q_1/2}]P$ first and performing a modular composition. Remark also that one can use an addition chain passing through $\{q_1/2, q_1, q_2\}$.

5.2 The case of isogeny cycles

The Abelian lift approach can be used to find $\lambda \bmod \ell^m$ in the isogeny cycle algorithm [8, 7]. Using the techniques described therein, we can compute a factor $f_{\ell^m,\lambda}(X)$ of the ℓ^m-division polynomial $f_{\ell^m}(X)$ (for some variants, this is a division polynomial for a curve E_m related to E).

Let φ denotes Euler's totient function, so that $\varphi(\ell^m) = \ell^{m-1}(\ell-1)$ for prime ℓ. Write $\lambda = c^x \bmod \ell^m$ with c a generator of $(\mathbb{Z}/\ell^m\mathbb{Z})^*$. Knowing $x \bmod (\ell-1)$ from the first computation, we show how to obtain $x \bmod \ell^{m-1}$ and therefore recover it modulo $\varphi(\ell^m)$ by Chinese Remaindering. To be quite general, as this is one of the advantages of the isogeny cycle approach, we suppose that $f_{\ell^m,\lambda}$ is a factor of degree $\delta\ell^{m-1}$ where δ is the *semi-order* of $\lambda \bmod \ell$ (if ϖ is the order, then the semi-order is ϖ if ϖ is odd and $\varpi/2$ if it is even). As a consequence $\delta \mid (\ell-1)/2$.

Let $x_0 = x \bmod (\ell-1)$, $q = \ell^{m-1}$ and $q' = (\ell-1)/2$. We let $h = c^{qx_0} \bmod \ell^m$ and $k = c^{q'} \bmod \ell^m$, where c generates $(\mathbb{Z}/\ell^m\mathbb{Z})^*$ and compute η_0 and η_1 as usual. The rest of the algorithm is unchanged. Adapting the analysis of Proposition 4.1 and writing $n = \delta\ell^{m-1}$, the complexity is

$$O(\mathcal{C}(n)\log(q') + \mathsf{M}(q)\log(p) + \mathcal{C}_{\sqrt{q}}(q)).$$

For $\ell \ll \log(p)$ (often the case in real-life examples), the dominant cost is $\mathsf{M}(q)\log(p)$. This is faster than the cost $O(\mathsf{M}(\delta q)\log(p))$ of computing Y^p by classical algorithms.

Numerical example. Consider the same curve $E : Y^2 = X^3 + X + 22$ over $\mathbb{F}_p$ with $p = 1009$. The eigenvalue 7 (resp. 3) has semi-order 6 (resp. 3), so we use $\lambda = 3$, as described in [7]. We get the following values, giving $\lambda = 3 \bmod 13^2$.

$$\begin{aligned}
f_{13^2,\lambda} &= X^{39} + 689X^{38} + 779X^{37} + 546X^{36} + 840X^{35} \\
&\quad +246X^{34} + 415X^{33} + 949X^{32} + 641X^{31} + 553X^{30} \\
&\quad +454X^{29} + 468X^{28} + 328X^{27} + 106X^{26} + 715X^{25} \\
&\quad +322X^{24} + 669X^{23} + 668X^{22} + 108X^{21} + 392X^{20} \\
&\quad +717X^{19} + 590X^{18} + 769X^{17} + 811X^{16} + 506X^{15} \\
&\quad +965X^{14} + 833X^{13} + 717X^{12} + 209X^{11} + 835X^{10} \\
&\quad +690X^9 + 938X^8 + 418X^7 + 670X^6 + 744X^5 \\
&\quad +29X^4 + 146X^3 + 914X^2 + 108X + 999, \\
\eta_0 &= 933\theta^{38} + 84\theta^{37} + 829\theta^{36} + 660\theta^{35} + 179\theta^{34} + 974\theta^{33} \\
&\quad +1878\theta^{32} + 581\theta^{31} + 773\theta^{30} + 84\theta^{29} + 227\theta^{28} \\
&\quad +631\theta^{27} + 938\theta^{26} + 852\theta^{25} + 962\theta^{24} + 153\theta^{23} \\
&\quad +969\theta^{22} + 128\theta^{21} + 588\theta^{20} + 670\theta^{19} + 120\theta^{18} \\
&\quad +52\theta^{17} + 906\theta^{16} + 654\theta^{15} + 934\theta^{14} + 897\theta^{13} \\
&\quad +966\theta^{12} + 827\theta^{11} + 569\theta^{10} + 864\theta^9 + 234\theta^8 + 671\theta^7 \\
&\quad +257\theta^6 + 160\theta^5 + 596\theta^4 + 995\theta^3 + 849\theta^2 + 548\theta \\
&\quad +228, \\
M &= T^{13} + 369T^{12} + 34T^{11} + 617T^{10} + 139T^9 + 579T^8 \\
&\quad +702T^7 471T^6 + 490T^5 + 740T^4 + 122T^3 + 271T^2 \\
&\quad +94T + 8, \\
\eta_1 &= 564\theta^{38} + 986\theta^{37} + 952\theta^{36} + 146\theta^{35} + 135\theta^{34} \\
&\quad +589\theta^{33} + 960\theta^{32} + 368\theta^{31} + 544\theta^{30} + 662\theta^{29} \\
&\quad +142\theta^{28} + 278\theta^{27} + 894\theta^{26} + 610\theta^{25} + 29\theta^{24} \\
&\quad +852\theta^{23} + 433\theta^{22} + 305\theta^{21} + 197\theta^{20} + 380\theta^{19} \\
&\quad +713\theta^{18} + 595\theta^{17} + 760\theta^{16} + 268\theta^{15} + 834\theta^{14} \\
&\quad +587\theta^{13} + 444\theta^{12} + 153\theta^{11} + 846\theta^{10} + 87\theta^9 \\
&\quad +578\theta^8 + 975\theta^7 + 512\theta^6 + 533\theta^5 + 321\theta^4 \\
&\quad +315\theta^3 + 828\theta^2 + 683\theta + 457 \\
C &= 405T^{12} + 652T^{11} + 538T^{10} + 407T^9 + 679T^8 \\
&\quad +796T^7 + 890T^6 + 497T^5 + 339T^4 + 240T^3 \\
&\quad +441T^2 + 924T + 420, \\
T_p &= 627T^{12} + 385T^{11} + 421T^{10} + 709T^9 + 117T^8 \\
&\quad +84T^7 + 911T^6 + 392T^5 + 97T^4 + 842T^3 \\
&\quad +646T^2 + 143T + 935
\end{aligned}$$

6. TIMINGS

We implemented all the algorithms using Shoup's NTL library [30]. Let $p = 10^{2499} + 7131$ (the record size so far) and $\ell = 5861$, so that $\ell - 1 = 2^2 \cdot 5 \cdot 293$. We have the following timings in seconds (all measures are taken on an AMD 64 Processor 3400+ at 2.4GHz).

q	η_0	η_1	$M(T)$	$C(T)$	T^p	v
4	15418	732	13	100	2	0
5	8491	446	17	43	10	0
293	3615	446	160	2509	3203	250

The total time is 36800 sec. Using the algorithm of [16], computing $Y^p \bmod f_{\ell,\lambda}$ costs 33001 sec, recovering X^p from Y^p takes 898 sec; concluding by finding λ takes 3650 sec.

Using the combination strategy, replacing $q = 4$ and $q = 5$ by $q = 20$, we find a total time of 23880 sec, which now outperforms the traditional approach.

q	η_0	η_1	$M(T)$	$C(T)$	T^p	v
20	11374	719	26	160	36	0

We next give an example of testing all possible combinations for $\ell = 421$, so that $\ell - 1 = 420 = 4 \times 3 \times 5 \times 7$. The minimal time corresponds to the combination 20×21. Indeed, when $\ell \ll \log(p)$ as is the case here, the dominant cost is that of computing T^p. This leads to selecting $\ell - 1 = q_1' q_2'$ with $q_1' \approx q_2'$. For comparison, the time for computing $Y^p \bmod f_{\ell,\lambda}(X)$ using classical approaches is 1855 sec, so that our new method is clearly superior in this case.

comb	η	M	C	T^p	v	Total
4×105	103.16	9.94	12.00	889.67	18.81	1066.61
12×35	105.02	5.67	8.39	328.43	1.07	489.57
20×21	100.86	4.91	8.00	132.21	0.04	287.75
28×15	94.45	4.59	7.62	188.67	0.22	343.93
60×7	88.80	4.66	7.35	308.35	1.65	459.48
84×5	91.03	4.94	7.36	483.28	2.91	625.08
140×3	86.51	5.74	7.89	881.06	7.42	1039.45

Figure 2 gives compares the *whole* Abelian Lift algorithm to the mere computation of Y^p in $\mathcal{A}$ for several values of ℓ; the column *fact* gives the factorization of $\ell - 1$, and the column *comb* the combination we used. As can be seen, our approach brings a significant improvement.

7. CONCLUSIONS

We have presented the implementation of a new method of computing the discrete logarithm step in the SEA algorithm for counting points on elliptic curves. The method applies in the Elkies variant of the algorithm; it computes the index of an eigenvalue $\lambda \in (\mathbb{Z}/\ell\mathbb{Z})^*$ in cyclic subgroups of the multiplicative groups separately and concludes by Chinese Remaindering. If q are the coprime divisors modulo which the index is computed, then the Frobenius will be evaluated in extensions of degree q of $\mathbb{F}_p$, an improvement with respect to one degree $(\ell-1)/2$ extension in the classical variant.

A further improvement can be achieved by using elliptic curve Gauss and Jacobi sums and the identity

$$\tau(\chi)^{p-\overline{p}_p} = \chi(\lambda)^{-p},$$

with the Gauss sum

$$\tau(\chi) \quad = \quad \textstyle\sum_{a=1}^{\ell-1} \chi(a)\,(g_a(\theta))$$

using the notations of Section 3. If χ is a character of order q, then the identity above yields the index $\log_\ell(\lambda) \bmod q$ and can be computed by means of Jacobi sums in an extension $\mathbb{F}_p[\xi]$ generated by a qth root of unity. The degree is thus once more reduced to $\mathrm{ord}_p(q)$. The method is described in [21] and is interesting for $\mathrm{ord}_p(q) \ll q$. The run time for exponentiation is reduced successively by the two variants; however this happens at the cost of estimation of traces. Any improvement in the trace algorithm, for instance by using modular functions, would be of importance.

Acknowledgments. Thanks to P. Gaudry for helpful discussions on polynomial arithmetic and for suggesting the combination approach of Section 5.1; to E. Kaltofen on trace algorithms. Thanks also to A. Bostan and A. Enge for some

ℓ	fact	comb	Total	Y^p
61	$4 \times 3 \times 5$	12×5	29.5858	242.991
229	$4 \times 3 \times 19$	12×19	161.262	952.416
241	$16 \times 3 \times 5$	16×15	134.288	1011.21
277	$4 \times 3 \times 23$	12×23	325.378	1463.9
281	$8 \times 5 \times 7$	8×35	443.064	1434.83
313	$8 \times 3 \times 13$	24×13	271.501	1468.8
337	$16 \times 3 \times 7$	16×21	239.125	1569.6
349	$4 \times 3 \times 29$	12×29	364.495	1587.61
397	$4 \times 9 \times 11$	36×11	394.333	1764.68
409	$8 \times 3 \times 17$	24×17	353.879	1825.81
421	$4 \times 3 \times 5 \times 7$	12×35	513.213	1855.14
457	$8 \times 3 \times 19$	24×19	391.897	2022.29
461	$4 \times 5 \times 23$	20×23	415.302	1998.82
521	$8 \times 5 \times 13$	40×13	512.104	2600.99
541	$4 \times 27 \times 5$	20×27	541.541	2767.78
613	$4 \times 9 \times 17$	36×17	620.348	3063.99
617	$8 \times 7 \times 11$	56×11	638.473	3023.47
673	$32 \times 3 \times 7$	32×21	618.209	3172.11
701	$4 \times 25 \times 7$	28×25	722.865	3257.17
709	$4 \times 3 \times 59$	12×59	948.196	3341.22
733	$4 \times 3 \times 61$	12×61	992.232	3374
881	$16 \times 5 \times 11$	16×55	1076.8	3911.75
937	$8 \times 9 \times 13$	72×13	1085.57	4175.14
953	$8 \times 7 \times 17$	56×17	980.541	4258.69
997	$4 \times 3 \times 83$	12×83	1547.85	4329.35
1033	$8 \times 3 \times 43$	24×43	1345.94	5484.26
1069	$4 \times 3 \times 89$	12×89	1822.46	5608.17
1093	$4 \times 3 \times 7 \times 13$	12×91	1849.94	5662.93
1213	$4 \times 3 \times 101$	12×101	2065.43	6063.6
1237	$4 \times 3 \times 103$	12×103	2103.97	6194.32
1277	$4 \times 11 \times 29$	44×29	1566.86	6321.76
1289	$8 \times 7 \times 23$	56×23	1622.27	6558
1361	$16 \times 5 \times 17$	80×17	1716.28	6634.66
1381	$4 \times 3 \times 5 \times 23$	12×115	2417.78	6683.73
1429	$4 \times 3 \times 7 \times 17$	12×119	2582.24	6839.37
1453	$4 \times 3 \times 121$	12×121	2591.88	6918.27
1481	$8 \times 5 \times 37$	40×37	2011.45	7150.25
1489	$16 \times 3 \times 31$	48×31	1855.47	7012.59
1549	$4 \times 9 \times 43$	36×43	2147.66	7346.68
1613	$4 \times 13 \times 31$	52×31	2118.97	7491.53
1657	$8 \times 9 \times 23$	72×23	2291.12	7730.41
1709	$4 \times 7 \times 61$	28×61	2488.44	7962.25
1741	$4 \times 3 \times 5 \times 29$	12×145	3580.26	8004.96
1753	$8 \times 3 \times 73$	24×73	2751.56	8091.46
1777	$16 \times 3 \times 37$	48×37	2471.74	8106.22
5861	$4 \times 5 \times 293$	20×293	22215.7	30641.1
5981	$4 \times 5 \times 13 \times 23$	20×299	22739.5	30551.3
8009	$8 \times 7 \times 11 \times 13$	56×143	30402.8	38457.1

Figure 2: Abelian lift vs. exponentiation

help. One of us (FM) would like to thank the Fields Institute for its hospitality during which part of this work was achieved. We are glad to have followed the suggestions of the referees that improved the presentation.

8. REFERENCES

[1] A. O. L. Atkin. The number of points on an elliptic curve modulo a prime (II). Available at `http://listserv.nodak.edu/archives/nmbrthry.html`, July 1992.

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

[3] A. Bostan, F. Morain, B. Salvy, and É. Schost. Fast algorithms for computing isogenies between elliptic curves, 2006.

[4] R. P. Brent and H. T. Kung. Fast algorithms for manipulating formal power series. *J. ACM*, 25(4):581–595, 1978.

[5] D. G. Cantor and E. Kaltofen. On fast multiplication of polynomials over arbitrary algebras. *Acta Informatica*, 28(7):693–701, 1991.

[6] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *J. Symb. Comput.*, 9(3):251–280, 1990.

[7] J.-M. Couveignes, L. Dewaghe, and F. Morain. Isogeny cycles and the Schoof-Elkies-Atkin algorithm. Research Report LIX/RR/96/03, LIX, Apr. 1996. Available at `http://www.lix.polytechnique.fr/Labo/Francois.Morain/`.

[8] J.-M. Couveignes and F. Morain. Schoof's algorithm and isogeny cycles. In L. Adleman and M.-D. Huang, editors, *Algorithmic Number Theory*, volume 877 of *Lecture Notes in Comput. Sci.*, pages 43–58. Springer-Verlag, 1994. 1st Algorithmic Number Theory Symposium - Cornell University, May 6-9, 1994.

[9] L. Dewaghe. *Calcul du nombre de points sur une courbe elliptique dans un corps fini*. Thèse, Université des Sciences et Technologies de Lille, Dec. 1996.

[10] L. Dewaghe. Remarks on the Schoof-Elkies-Atkin algorithm. *Math. Comp.*, 67(223):1247–1252, July 1998.

[11] N. D. Elkies. Explicit isogenies. Draft, 1992.

[12] N. D. Elkies. Elliptic and modular curves over finite fields and related computational issues. In *Computational Perspectives on Number Theory: Proceedings of a Conference in Honor of A. O. L. Atkin*, volume 7 of *AMS/IP Studies in Advanced Mathematics*, pages 21–76. AMS, International Press, 1998.

[13] A. Enge. Computing modular polynomials in quasi-linear time, 2006.

[14] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, 1999.

[15] J. von zur Gathen and V. Shoup. Computing Frobenius maps and factoring polynomials. *Comput. Complexity*, 2(3):187–224, 1992.

[16] P. Gaudry and F. Morain. Fast algorithms for computing the eigenvalue in the Schoof-Elkies-Atkin algorithm. In *ISSAC'06*, pages 109–115. ACM Press, 2006.

[17] E. Kaltofen and V. Shoup. Subquadratic-time factoring of polynomials over finite fields. *Math. Comput.*, 67(223):1179–1197, 1998.

[18] H. W. J. Lenstra. Galois theory and primality testing. In *Orders and their applications*, volume 1142 of *Lecture Notes in Mathematics*, pages 1–21. Springer Verlag, 1985.

[19] M. Maurer and V. Müller. Finding the eigenvalue in Elkies' algorithm. *Experiment. Math.*, 10(2):275–285, 2001.

[20] P. Mihăilescu. Cyclotomy primality proofs and their certificates. Mathematica Goettingensis, 2006.

[21] P. Mihăilescu. Elliptic curve Gauss sums and counting points. Mathematica Goettingensis, 2006.

[22] F. Morain. Calcul du nombre de points sur une courbe elliptique dans un corps fini : aspects algorithmiques. *J. Théor. Nombres Bordeaux*, 7:255–282, 1995.

[23] C. Pascal and É. Schost. Change of order for bivariate triangular sets. In *ISSAC'06*, pages 277–284. ACM Press, 2006.

[24] F. Rouillier. Solving zero-dimensional systems through the Rational Univariate Representation. *Appl. Alg. in Eng. Comm. Comput.*, 9(5):433–461, 1999.

[25] A. Schönhage and V. Strassen. Schnelle Multiplikation großer Zahlen. *Computing*, 7:281–292, 1971.

[26] R. Schoof. Elliptic curves over finite fields and the computation of square roots mod p. *Math. Comp.*, 44(170):483–494, 1985.

[27] R. Schoof. Counting points on elliptic curves over finite fields. *J. Théor. Nombres Bordeaux*, 7:219–254, 1995.

[28] J.-P. Serre. Propriétés galoisiennes des points d'ordre fini des courbes elliptiques. *Invent. Math.*, 15(4):259–331, 1971.

[29] V. Shoup. Fast construction of irreducible polynomials over finite fields. *J. Symbolic Comput.*, 17:371–391, 1994.

[30] V. Shoup. A new polynomial factorization algorithm and its implementation. *J. Symbolic Comput.*, 20:363–397, 1995.

[31] V. Shoup. Efficient computation of minimal polynomials in algebraic extensions of finite fields. In *ISSAC'99*, pages 53–58. ACM Press, 1999.