Computing isogenies between elliptic curves over F_{p^n} using Couveignes's algorithm. Mathematics of Computation, 69(229):351–370, January 2000.

COMPUTING ISOGENIES BETWEEN ELLIPTIC CURVES OVER F_{p^n} USING COUVEIGNES'S ALGORITHM

REYNALD LERCIER AND FRANÇOIS MORAIN

Abstract. The heart of the improvements of Elkies to Schoof's algorithm for computing the cardinality of elliptic curves over a finite field is the ability to compute isogenies between curves. Elkies' approach is well suited for the case where the characteristic of the field is large. Couveignes showed how to compute isogenies in small characteristic. The aim of this paper is to describe the first successful implementation of Couveignes's algorithm. In particular, we describe the use of fast algorithms for performing incremental operations on series. We also insist on the particular case of the characteristic 2.

1. Introduction

Elliptic curves have been used successfully to factor integers [27, 39], and prove the primality of large integers [5, 20, 3]. Moreover they turned out to be an interesting alternative to the use of $\mathbb{Z}/N\mathbb{Z}$ or finite fields in cryptographical schemes (see [38, 25], [36] and the survey in [30]).

One of the main algorithmic problems to be solved is the efficient computation of the cardinality of elliptic curves over finite fields. It was not until recently that Schoof's polynomial time algorithm for solving this problem could be efficiently used, due to the work of Atkin [1, 2] and Elkies [16, 17] (see also [44, 40] and the results of the implementations given in [40, 30, 26, 42]). The main ingredient is the use of explicit isogenies between elliptic curves. The methods developed there gave satisfactory results in the large characteristic case, but could not be used when the characteristic is small, which explains why the implementation of [37] did not give satisfactory results. The first solution to this problem was given in Couveignes's thesis [9].

The aim of this paper is to explain how Couveignes's algorithm can be implemented in an efficient way. The structure is as follows. Section 2 recalls basic facts on elliptic curves. In section 3, the particular case of isogenies of degree p will be treated, which will yield properties of the multiplication by p on elliptic curves; we will also deduce from these an algorithm for computing a factor of the p^e-division polynomial. Properties of the formal group will be presented in section 4. Section 5 will explain the decisive ideas of Couveignes for the computation of isogenies in small characteristic. Section 6 is concerned with fast algorithms for incremental computations on series. Section 7 details the algorithms we need to implement Couveignes's ideas. The complexity of Couveignes's approach is then derived. Section 8 will be devoted to the implementation in the special case of the characteristic 2.

To simplify the exposition, we will consider non-supersingular elliptic curves only. (Note that this is enough for the application to point counting, the cardinality of supersingular curves being studied in [35, 41].)

2. Preliminaries and notations

Throughout the paper, we let $\mathbb{K} = \mathbb{F}_q = \mathbb{F}_{p^n}$ be a finite field of characteristic p and denote by $\overline{\mathbb{K}}$ its Galois closure. The norm of an element x of $\mathbb{K}$ is written $\mathrm{N}_{\mathbb{K}/\mathbb{F}_p}(x)$ and the trace is noted $\mathrm{Tr}_{\mathbb{K}/\mathbb{F}_p}(x)$.

We will encounter many p-th roots in characteristic p and it will be convenient to write them as $\tilde{a} = \sqrt[p]{a}$ (note that every element a in $\mathbb{K}$ has exactly one p-th root given by $a^{p^{n-1}}$). Moreover, if $A(X)$ is a series (or a polynomial) in $\mathbb{K}[X]$: $A(X) = \sum_i a_i X^i$, we will write $\tilde{A}(X) = \sum_i \tilde{a}_i X^i$.

Date: January 10, 2015.

1991 *Mathematics Subject Classification.* Primary 11G20; Secondary 11T71, 11Y16, 94A60.

Key words and phrases. Elliptic curves, finite fields, isogenies, formal groups, Schoof's algorithm.

The second author is on leave from the French Department of Defense, Délégation Générale pour l'Armement.

As far as time complexity is concerned, our unit of cost will be the time needed to perform a multiplication in $\mathbb{K}$, a unit being thus $O(n^2(\log p)^2)$ bit complexity if ordinary multiplication is used. The space unit will be that of storing an element of $\mathbb{K}$, that is $O(n \log p)$ bits.

We recall well known properties of elliptic curves. All these can be found in [45]. Let E be an elliptic curve defined over $\mathbb{K}$ with defining equation $\mathfrak{E}(X, Y) = 0$ where

$$\mathfrak{E}(X, Y) = Y^2 + a_1XY + a_3Y - (X^3 + a_2X^2 + a_4X + a_6).$$

The curve E will be abbreviated as $[a_1, a_3, a_2, a_4, a_6]$. Remember that by Hasse's theorem, one has $\#E(\mathbb{K}) = q + 1 - c$ for some integer c, $|c| \leq 2\sqrt{q}$.

For an integer m, there exists polynomials ϕ_m, ψ_m and ω_m in $\mathbb{Z}[a_1, a_2, a_3, a_4, a_6, X, Y]$ such that

$$[m](X, Y) = \left(\frac{\phi_m(X, Y)}{\psi_m^2(X, Y)}, \frac{\omega_m(X, Y)}{\psi_m^3(X, Y)}\right).$$

A particular role is played by the polynomial ψ_m. We let $\psi'_m(X)$ denote $\psi_m(X, Y)$ mod $\mathfrak{E}(X, Y)$. When m is even we let $f_m = \psi'_m/(2Y + a_1X + a_3)$ and if m is odd, then $f_m = \psi'_m$. The m-torsion points of E, noted $E[m] = \{P \in E(\overline{\mathbb{K}}), mP = O_E\}$ can be described using $f_m(X)$: if P is a point on $E(\overline{\mathbb{K}})$ such that $2P \neq O_E$, then $P \in E[m]$ if and only if $f_m(X) = 0$.

3. Isogenies of degree p

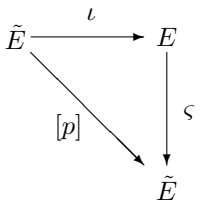
The aim of this section is to explain the properties of isogenies of degree p in characteristic p and to deduce from these results about the multiplication by p on E. These results will be used in the following section and will be a key to simplifying some parts of the subsequent algorithms. We are indebted to J.-M. Couveignes for the following facts.

Let $\Phi_\ell(X, Y)$ denote the ℓ-th modular polynomial, that is the polynomial for which the roots of $\Phi_\ell(X, j(E))$ are the j-invariants of the elliptic curves E^* related to E by an isogeny of degree ℓ. A theorem of Kronecker tells us (see for instance [4]) that

$$\Phi_p(X, Y) \equiv (X^p - Y)(X - Y^p) \bmod p.$$

This immediately shows that if E and E^* are p-isogenous, then $j(E^*) = j(E)^p$ or $j(E^*) = j(E)^{1/p}$.

If $E = [a_1, a_3, a_2, a_4, a_6]$, then define $E^p = [a_1^p, a_3^p, a_2^p, a_4^p, a_6^p]$ and $\tilde{E} = [\tilde{a}_1, \tilde{a}_3, \tilde{a}_2, \tilde{a}_4, \tilde{a}_6]$. Let us look at the following diagram:



Multiplication by p on $\tilde{E}$ factors as a product of isogenies $[p] = \varsigma \circ \iota$ where $\iota : (X, Y) \mapsto (X^p, Y^p)$ is inseparable and ς is separable. We can reformulate this as:

Proposition 3.1. *Multiplication by p on E is given by*

$$[p](X, Y) = (F_p(X)^p, G_p(X, Y)^p) \tag{1}$$

where $F_p(X)$ and $G_p(X, Y)$ are two rational fractions.

As a useful corollary, we note

Corollary 3.1. *There exists a polynomial $\mathfrak{f}_{p^e} \in \mathbb{K}[X]$ such that the division polynomial f_{p^e} can be written as $f_{p^e}(X) = (\mathfrak{f}_{p^e}(X))^{p^e}$.*

Proof: From (1), it follows that

$$[p^e](X, Y) = (F_{p^e}(X)^{p^e}, G_{p^e}(X, Y)^{p^e})$$

for all $e \geq 1$. In particular, this implies that ψ_{p^e} and *a fortiori* f_{p^e} are p^e-th powers. From this it follows that the degree of f_{p^e} is in fact at most $(p^{2e} - p^e)/2$. □

Remark. As shown in [31], there is an elementary approach to the fact that $f_p(X)$ is a p-th power in characteristic p, using Fricke's differential equation [18, vol II, pp. 191].

Corollary 3.2. *For $e > 1$, $\mathfrak{f}_{p^{e-1}}(X) \mid \mathfrak{f}_{p^e}(X)$ and the resulting quotient is a factor of f_{p^e} of degree $p^{e-1}(p-1)/2$ if p is odd and 2^{e-2} if $p = 2$.*

In practice, the computation of $\mathfrak{f}_p$ is done using [37] for $p = 2$ and [21] when p is odd. Then $\mathfrak{f}_{p^e}$ is computed using the isogenies given by Vélu's formulae as in [15] and the methods of [13, 12].

4. Formal groups

The material below is taken from [45, Chap. IV][1]. Let $t = -X/Y$ and $s = -1/Y$. We transform the equation of E to get:

$$s = A(t,s) = t^3 + a_1ts + a_2t^2s + a_3s^2 + a_4ts^2 + a_6s^3. \tag{2}$$

Substituting this equation into itself, we get s as a power series in t, that we will note $\mathcal{S}(t)$. Since equation (2) is again cubic, we can add two points $(t_1, s(t_1))$ and $(t_2, s(t_2))$ to get $(t_3, s(t_3))$ in the usual way, using the tangent-and-chord law. As a result, we find t_3 as a power series $F_a(t_1,t_2)$ in $\mathbb{K}[[t_1,t_2]]$ whose first terms are:

$$F_a(t_1,t_2) = t_1 + t_2 - a_1t_1t_2 - a_2(t_1^2t_2 + t_1t_2^2) - (2a_3t_1^3t_2 - (a_1a_2 - 3a_3)t_1^2t_2^2 + 2a_3t_1t_2^3) + \cdots \tag{3}$$

This series F_a defines what is known as the *formal group* $\mathcal{E}$ associated to E.

4.1. **Computing $\mathcal{S}(t)$.** From the equation $\mathcal{S} = A(t,\mathcal{S})$, it is easy to compute the first L coefficients of $\mathcal{S}$ as a formal series in t, by an iterative process in $O(L^2)$ operations. We can do better using standard techniques from combinatorics [8, 43]. In particular, $\mathcal{S}$ satisfies a second order linear differential equation with polynomial coefficients in t, from which we can easily deduce recurrence relations between the coefficients of $\mathcal{S}$. Hence, these coefficients can be computed in $O(L)$ operations modulo precomputations. See section 8.1 for the computations in characteristic 2.

The first coefficients of $\mathcal{S}$ are:

$$\mathcal{S}(t) = \sum_{i=3}^{\infty} s_it^i = t^3 + a_1\,t^4 + (a_1^2 + a_2)t^5 + O(t^6). \tag{4}$$

Given any $t(\tau)$ in $\mathcal{E}$, we can compute $\mathcal{S}(t(\tau))$ by the same algorithms.

We deduce from this that

$$Y = -\frac{1}{s} = -t^{-3} + a_1t^{-2} + a_2t^{-1} + a_3 + (a_1a_3 + a_4)t + O(t^2),$$

and

$$X = \frac{t}{s} = -tY = t^{-2} - a_1t^{-1} - a_2 - a_3t - (a_1a_3 + a_4)t^2 + O(t^3),$$

$$Z = \frac{1}{X} = t^2 + a_1t^3 + (a_1^2 + a_2)t^4 + O(t^5).$$

4.2. **Group law.** In this section, we give the formulas that will be used for computing the group law on $\mathcal{E}$, that we will note $\oplus$, multiplication by k being noting as $[k]$. The neutral element is $O_{\mathcal{E}} = (0,0)$ and the equation of $\mathcal{E}$ is $F(t,s) := A(t,s) - s = 0$. Proofs can be found in the reference.

We start from two points $P_1 = (t_1,s_1)$ and $P_2 = (t_2,s_2)$, different from $O_{\mathcal{E}}$, and we compute the sum $(t_3,s_3) = (t_1,s_1) \oplus (t_2,s_2)$. This is done as usual: the line passing through P_1 and P_2 intersects $\mathcal{E}$ in a third point $P_i = (t_i,s_i)$ and the line passing through P_i and $(0,0)$ intersects $\mathcal{E}$ at P_3.

We let $y = \lambda t + \nu$ be the line passing through the two points P_1 and P_2. If $(t_1,s_1) \neq (t_2,s_2)$ then

$$\lambda = \frac{s_2 - s_1}{t_2 - t_1} = t_1^2 + t_1t_2 + t_2^2 + \cdots \tag{5}$$

and if the two points are equal

$$\lambda = -\frac{\dfrac{\partial F}{\partial t}}{\dfrac{\partial F}{\partial s}} = -\frac{a_1\,s_1 + 3\,t_1^2 + 2\,a_2\,t_1s_1 + a_4\,s_1^2}{-1 + a_1\,t_1 + 2\,a_3\,s_1 + a_2\,t_1^2 + 2\,a_4\,t_1s_1 + 3\,a_6\,s_1^2} = 3t_1^2 + 4a_1t_1^3 + O(t_1^4). \tag{6}$$

[1] Be careful that there are some typos and missing equations in [23, Chap. 12].

One computes also $\nu = s_1 - \lambda t_1$. For P_i, one finds

$$t_i = -t_1 - t_2 - \frac{a_1\,\lambda + 2\,a_4\,\nu\,\lambda + a_2\,\nu + 3\,a_6\,\nu\,\lambda^2 + a_3\,\lambda^2}{a_4\,\lambda^2 + a_2\,\lambda + 1 + a_6\,\lambda^3} \tag{7}$$

and from this, we deduce $s_i = \lambda t_i + \nu$.

If $t_i = 0$, then P_2 is the opposite of P_1 and we are done, $P_3 = O_{\mathcal{E}}$. Otherwise, we have to compute the addition of (t_i, s_i) and the origin point $(0, 0)$ to get (t_3, s_3). Finally we obtain

$$t_3 = \frac{t_i}{-1 + a_1\,t_i + a_3\,s_i},\ s_3 = \frac{s_i}{t_i}\,t_3. \tag{8}$$

from which we recover $t_3 = F_a(t_1, t_2)$.

It is now easy to compute the opposite of P, simply noting that this opposite is the third point of intersection of the line joining P and $O_{\mathcal{E}}$ with $\mathcal{E}$. Precisely, if $-P = (t', s')$, one has

$$t' = \frac{t}{-1 + a_1 t + a_3 s}. \tag{9}$$

When $t_1 = t_2$, we get

$$F_d(t_1) = [2]t_1 = 2\,t_1 - a_1\,t_1^2 - 2\,a_2\,t_1^3 + (a_1\,a_2 - 7\,a_3)\,t_1^4 + O(t_1^5). \tag{10}$$

4.3. **The Hasse invariant.** Using Proposition 1, we see that

$$[p](t) = -\left(\frac{F_p(t/\mathcal{S}(t))}{G_p(t/\mathcal{S}(t), -1/\mathcal{S}(t))}\right)^p, \mathcal{S}([p](t)) = -\left(\frac{1}{G_p(t/\mathcal{S}(t), -1/\mathcal{S}(t))}\right)^p. \tag{11}$$

Let us introduce the rational fraction

$$\mathcal{R}_{p,E}(t, s) = (-1)^p \frac{F_p(t/s)}{G_p(t/s, -1/s)} \tag{12}$$

and put $\Psi_{p,E}(t) = \mathcal{R}_{p,E}(t, \mathcal{S}(t))$. Then $[p](t) = \Psi_{p,E}(t)^p$. We will see the interest of such a definition in section 7.2.2.

Example. If $p = 2$ and $E = [1, 0, 0, 0, a_6]$, one has

$$\mathcal{R}_{2,E}(t, s) = \frac{\left(t^2 + \tilde{a}_6 s^2\right) t}{t^3 + (1 + \tilde{a}_6 s)\,t^2 + (\tilde{a}_6 s^2 + s)\,t + \tilde{a}_6 s^2 + s}. \tag{13}$$

Theorem 4.1. *Assume E is not supersingular. Then*

$$[p](t) = c_p(E) t^p + O(t^{p^2})$$

where the coefficient $c_p(E)$ of t^p is called the (relative) Hasse invariant *of E.*

One of the important property of the Hasse invariant is the following [45, Chap. V, §4]:

Theorem 4.2. *Let $\#E(\mathbb{K}) = q + 1 - c$. The Hasse invariant satisfies: $\mathrm{N}_{\mathbb{K}/\mathbb{F}_p}(c_p(E)) \equiv c \bmod p$.*

Remembering that two isogenous curves have the same number of points (see [14, 46]), the preceding theorem tells us the following:

Corollary 4.1. *Two isogenous curves E and E^* have Hasse invariants related by*

$$c_p(E^*) \equiv \varepsilon^{p-1} c_p(E) \bmod p$$

for some ε in $\mathbb{K}^*$.

In characteristic 2, one has $c_p(E) = a_1$; in characteristic 3, for the curve $[0, 0, a_2, 0, a_6]$, it is a_2. When $p > 3$, one can compute the Hasse invariant using the work of Deuring [14] or Atkin's method using hypergeometric polynomials (see [2, 24]).

5. Couveignes's algorithm: the theory

5.1. **An overview.** Let E and E^* be two elliptic curves defined over $\mathbb{K}$ by

$$\begin{array}{rrcl} E: & Y^2+a_1XY+a_3Y & = & X^3+a_2X^2+a_4X+a_6, \\ E^*: & Y^2+a_1^*XY+a_3^*Y & = & X^3+a_2^*X^2+a_4^*X+a_6^*, \end{array}$$

such that there exists an isogeny $\mathfrak{I}$ of degree ℓ between them. In view of corollary 4.1, we can assume without loss of generality that $c_p(E^*)=c_p(E)=\gamma$.

We assume that the isogeny $\mathfrak{I}$ between E and E^* is given by

$$\begin{array}{rccc} \mathfrak{I}: & E & \longrightarrow & E^* \\ & (X,Y) & \longmapsto & \left(\dfrac{g(X)}{h^2(X)}, \dfrac{r(X)+Yt(X)}{h^3(X)}\right), \end{array}$$

where $g(X)$, $h(X)$, $r(X)$ and $t(X)$ are polynomials of degree ℓ, $(\ell-1)/2$, $3(\ell+1)/2$ and $3(\ell-1)/2$. The aim of Couveignes's algorithm [9] is the computation of $g(X)$ and $h(X)$ given the equations of E and E^*.

Let $I(X)=g(X)/h^2(X)$. It is equivalent to search for g and h such that

$$I: X \mapsto X^* = I(X) = \frac{g(X)}{h(X)^2}$$

or for $\hat{I}$ which sends $Z=1/X$ to $Z^*=1/X^*$, that is

$$\hat{I}: Z \mapsto Z^* = \hat{I}(Z) = Z\frac{\hat{h}^2(Z)}{\hat{g}(Z)}$$

with $\hat{g}(Z)=Z^\ell g(1/Z)$ and $\hat{h}(Z)=Z^{(\ell-1)/2}h(Z)$. We note that $\hat{g}$ has degree ℓ. It is well known that the coefficients of the expansion of a rational fraction $F(Z)$ with denominator of degree ℓ around $Z=0$ satisfy a recurrence relation of depth ℓ (see section 7.3.2 for more details). Reciprocally, given the 2ℓ first coefficients, one can recover $F(Z)$ exactly. Couveignes's idea is just this: finding a series that looks like an isogeny and then check whether it comes from a fraction whose denominator has degree ℓ. In fact, we compute $2\ell+2$ terms of the series, thus obtaining in general a fraction with denominator of degree *a priori* $\ell+1$. If this denominator turns out to have degree ℓ, then we are almost sure to have the correct value for $\hat{I}$. See section 7.3 for more details.

Enumerating the putative isogenies is possible using the formal groups associated to E and E^* as described below.

5.2. **Morphisms of formal groups.** As shown in section 2, associated to E and E^*, there are two formal groups $\mathcal{E}$ and $\mathcal{E}^*$,

$$\begin{array}{rrcl} \mathcal{E}: & t^3+a_1ts+a_2t^2s+a_3s^2+a_4ts^2+a_6s^3-s & = & 0, \\ \mathcal{E}^*: & t^3+a_1^*ts+a_2^*t^2s+a_3^*s^2+a_4^*ts^2+a_6^*s^3-s & = & 0. \end{array}$$

A *morphism* of formal groups is given by $\mathcal{M}$ such that for all formal points $(t_1(\tau),s_1(\tau))$ and $(t_2(\tau),s_2(\tau))$ of $\mathcal{E}$:

$$\mathcal{M}((t_1(\tau),s_1(\tau)) \oplus (t_2(\tau),s_2(\tau))) = \mathcal{M}((t_1(\tau),s_1(\tau))) \oplus \mathcal{M}((t_2(\tau),s_2(\tau))).$$

Associated to a morphism $\mathcal{M}$ between $\mathcal{E}$ and $\mathcal{E}^*$, there is a series

$$\mathcal{U}(t) = \sum_{i\geq 1} u_i t^i,$$

such that a point $(t(\tau),s(\tau))$ of $\mathcal{E}$ is sent to the point $\mathcal{M}(t(\tau),s(\tau)) = (\mathcal{U}(t(\tau)), \mathcal{S}^*(\mathcal{U}(t(\tau))))$ of $\mathcal{E}^*$ ($\mathcal{S}^*$ is defined by (4)). *A fortiori*, the series $\mathcal{U}(t)$ satisfies

$$\mathcal{U}(t_1 \oplus t_2(\tau)) = \mathcal{U}(t_1(\tau)) \oplus \mathcal{U}(t_2(\tau)) \tag{14}$$

from which $\mathcal{U}\circ[n]=[n]\circ\mathcal{U}$ for any integer n.

Coming back to our problem, $\hat{I}$ gives rise to a morphism $\mathcal{I}$ between $\mathcal{E}$ and $\mathcal{E}^*$, and to a series $\mathcal{W}$. The problem is now the following: among all morphisms between $\mathcal{E}$ and $\mathcal{E}^*$, determine which is the one coming from $\mathcal{I}$, or equivalently, among all series satisfying (14), determine which is the one coming from $\hat{I}$. Since the set of morphisms from $\mathcal{E}$ to $\mathcal{E}^*$ is a $\mathbb{Z}_p$-module of rank 1 (see [19]), we run through all the powers of a generator of this module and test for each morphism whether we can recover $\hat{I}$. Our first task is then to find a generator of this module.

Since $2\ell+2$ terms of $\hat{I}(Z)$ are needed, and since $Z = 1/X = s/t = t^2 + O(t^3)$, this means that we need $\mathcal{L} = 4\ell + 2$ terms of the series $\mathcal{W}$ associated to $\mathcal{I}$. In other words, we need to consider a finite number of series in order to find the good one. We will compute the precise number of such series in the following section.

5.3. **Finding conditions satisfied by morphisms.** Let us now look at the properties satisfied by morphisms between $\mathcal{E}$ and $\mathcal{E}^*$, or more precisely by the associated series. We will compute the first $\mathcal{L}$ coefficients of

$$\mathcal{U}(t) = \sum_{i=1}^{\infty} u_i t^i$$

by induction. Let us assume that $u_1, \ldots, u_{i-1}$ are known. An ingenious exploitation of equation (14) will allow us to calculate u_i.

Let us specialize $t_1(\tau) = \tau$ and $t_2(\tau) = A\tau$ where A is in $\mathbb{K}$. Considering the left hand side of $\mathcal{U}(\tau \oplus A\tau) = \mathcal{U}(\tau) \oplus \mathcal{U}(A\tau)$, we find that u_i appears alone in the coefficient of τ^i as $(1+A)^i u_i$ among terms depending only on $u_1, u_2, \ldots, u_{i-1}$. On the other hand,

$$\mathcal{U}(\tau) \oplus \mathcal{U}(A\tau) = \mathcal{U}(\tau) + \mathcal{U}(A\tau) + P(\mathcal{U}(\tau), \mathcal{U}(A\tau))$$

where $P(\mathcal{U}(\tau), \mathcal{U}(A\tau))$ contains monomials of total degree greater than 1 in $\mathcal{U}(\tau)$ and $\mathcal{U}(A\tau)$. This means that u_i appears in the coefficient of τ^i as $(1 + A^i)u_i$ among terms depending only on $u_1, u_2, \ldots, u_{i-1}$. From this, we deduce that

$$u_i \left((1+A)^i - 1 - A^i\right) + e_i(A, u_1 \ldots, u_{i-1}) = 0, \tag{15}$$

with e_i a multivariate polynomial. If $(1+A)^i \neq 1 + A^i$, this relation gives us u_i. We see that this condition on A cannot be met when i is a power of p, but for other values of i, we can find A such that it is realized, at least if $i < q$.

Suppose now that $i = p^e$. Exploiting the equation $\mathcal{U}([p]\tau) = [p](\mathcal{U}(\tau))$, we find that u_i satisfies:

$$\left(\frac{u_i}{\eta}\right) - \left(\frac{u_i}{\eta}\right)^p = f_i(u_1, \ldots, u_{i-1}) \tag{16}$$

where

$$\eta = \gamma^{(p^e-1)/(p-1)}$$

and f_i is a multivariate polynomial. We will see in section 7.1.2 how to solve this equation. Obviously, it has at most p solutions.

Let us look at the case $i = 1 = p^0$. The corresponding equation is simply $u_1^{p-1} = 1$. Therefore, u_1 is in the prime field and w.l.o.g. we can take $u_1 = 1$.

5.4. **Enumerating all morphisms.** We can summarize the results of the preceding section as follows. Once u_{p^e} is fixed, all coefficients u_j for $p^e < j < p^{e+1}$ are uniquely determined. In this way, we can count the number of different truncated morphisms up to order $\mathcal{L}$. Let $p^r < \mathcal{L} < p^{r+1}$. Then there are at most p^{r+1} distinct series. For each e, $1 \leq e \leq r$, there are at most p values for u_{p^e}; if $e = 0$ this number is at most $p-1$ since $u_1 = 0$ is not valid. Therefore, there are $p^r(p-1)$ morphisms $\mathcal{U}$. We need to enumerate them in order to find the one that comes from an isogeny.

5.4.1. *First approach.* It consists in testing all possible values of u_{p^e} for each e, using a backtracking procedure, that is straightforward from the explanations given above.

5.4.2. *Second approach.* Let $\mathcal{U}$ be any generator of the set of morphisms between $\mathcal{E}$ and $\mathcal{E}^*$, found as in the preceding section. There exists a p-adic integer N such that $\mathcal{W} = [N] \circ \mathcal{U}$. Write

$$N = \sum_{i=0}^{\infty} n_i p^i.$$

Remembering that $p^r < \mathcal{L} < p^{r+1}$, we write

$$[N] \circ \mathcal{U} = \bigoplus_{i=0}^{r} \left([n_i] \circ ([p^i] \circ \mathcal{U})\right) \oplus \bigoplus_{i>r} \left([n_i] \circ ([p^i] \circ \mathcal{U})\right).$$

But the valuation of the series $[p^i](t)$ is p^i, which implies that, when $i > r$, the terms coming from $[p^i] \circ \mathcal{U}$ do not provide any contribution to the first $\mathcal{L}$ coefficients of $[N] \circ \mathcal{U}$. So, it is enough to check whether one of the series $[N] \circ \mathcal{U}$ comes from an isogeny for $N < p^{r+1}$. Moreover, n_0 cannot be 0.

We can reduce the number of tentative morphisms, using the following result.

Proposition 5.1. *Let N be an integer satisfying $0 \leq N < p^{r+1}$. Then one has*

$$([p^{r+1} - N] \circ \mathcal{U})(t) = ([-N] \circ \mathcal{U})(t) + O(t^{p^{r+1}}). \tag{17}$$

Proof: The result follows easily from (4.1). □

This relation expresses the fact that the morphism $-\mathcal{W}$ associated with the isogeny $-\hat{I}$, has the same abscissa as $\hat{I}$. So, at least one morphism $\mathcal{M} = [N] \circ \mathcal{U}$ for $N < p^{r+1}/2$ and N prime to p is equal to $\mathcal{W}$ or $-\mathcal{W}$ and is associated to the abscissa of $\hat{I}$. That is to say, we have to compute at most $p^r(p-1)/2$ morphisms $\mathcal{M}$.

6. Incremental series computations

The implementation of Couveignes's algorithm requires the use of fast algorithms for series computations. As will be described in section 7, the algorithms we need are concerned with incremental computations. Starting from series whose coefficients are known up to order i, we will find the terms of order $i+1$ of these series.

In these sections, we note for any series $\mathcal{A}(\tau) = \sum_{i \geq v}^{\infty} a_i \tau^i$ of valuation v in $\mathbb{K}[[\tau]]$, $\mathcal{A}(\tau)_k$ the finite sum $\sum_{i=v}^{k} a_i \tau^i$. The i-th coefficient of a series $\mathcal{A}$ will always denote a_i. We make the general assumption that multiplying two series with m terms uses $O(m^\mu)$ units; of course, we assume $1 < \mu \leq 2$.

Incremental algorithms for the four basic operations $+, -, \times, /$ are easy to derive and we will not give them.

6.1. Computations in the formal group.

Let $(\mathcal{V}(t), \mathcal{S}(\mathcal{V}(t)))$ be a formal point of $\mathcal{E}$. We note

$$\mathcal{V}(t) = \sum_{i=1}^{\infty} v_i t^i \text{ and } \mathcal{S}(\mathcal{V}(t)) = \sum_{i=3}^{\infty} \varpi_i t^i.$$

The following propositions summarize our approach:

Proposition 6.1. *We can obtain ϖ_L from $(\varpi_3, \ldots, \varpi_{L-1})$ and $(v_1, \ldots, v_{L-2})$ with $O(L)$ multiplications in $\mathbb{K}$.*

Proof: As $(\mathcal{V}(t), \mathcal{S}(\mathcal{V}(t)))$ is an element of the formal group defined by $\mathcal{E}$,

$$\mathcal{V}^3 + a_1 \mathcal{V}\mathcal{S}(\mathcal{V}) + a_2 \mathcal{V}^2 \mathcal{S}(\mathcal{V}) + a_3 \mathcal{S}(\mathcal{V})^2 + a_4 \mathcal{V}\mathcal{S}(\mathcal{W})^2 + a_6 \mathcal{S}(\mathcal{V})^3 = \mathcal{S}(\mathcal{V}). \tag{18}$$

By inspection of the valuations and indices of $\mathcal{V}$ and $\mathcal{S}$, the result follows. □

The same work can be done for the addition in $\mathcal{E}$:

Proposition 6.2. *Let $\mathcal{A}(t) = \sum_i a_i t^i$ and $\mathcal{A}'(t) = \sum_i a'_i t^i$ be two formal series and put $\mathcal{S}(t) = \mathcal{S}(\mathcal{A}(t)) = \sum_{i=3}^{\infty} \varpi_i t^i$ (resp. $\mathcal{S}'(t) = \mathcal{S}(\mathcal{A}'(t)) = \sum_{i=3}^{\infty} \varpi'_i t^i$). We can obtain the L-th coefficient of $(\mathcal{A}(t) \oplus \mathcal{A}'(t))_L$ from the truncated formal points $(\mathcal{A}(t)_L, \mathcal{S}_{L+1}(t))$ and $(\mathcal{A}'(t)_L, \mathcal{S}'_{L+1}(t))$ with $O(L)$ operations.*

Finally,

Proposition 6.3. *We can obtain the L-th term of the series $\mathcal{R}(t, \mathcal{S}(t)) \circ \mathcal{V}(t)$ from the truncated formal point $(\mathcal{V}(t)_L, \mathcal{S}(\mathcal{V}(t))_{L+1})$ using $O(L)$ operations.*

6.2. An incremental algorithm for composition of series.

Let $f(t) = \sum_{i=1}^{\infty} a_i t^i$ and $g(t) = \sum_{i=0}^{\infty} b_i t^i$ be two formal series in $\mathbb{K}[[t]]$. We want to compute the series

$$h(t) = (g \circ f)(t) = \sum_{i=0}^{\infty} c_i t^i$$

incrementally. More precisely, we assume f is known up to order L and that we need to compute the coefficients $h_0, h_1, \ldots, h_L$ one at a time, or equivalently, given all series at order i, find h_i. We do this by an incremental version of the algorithm of Brent and Kung [7].

Let B be an integer $\leq L$ that we will determine later on. Let i be an integer less than L and assume we know all coefficients of g (resp. h) of index $< i$. We are looking for c_i. To compute $g_i \circ f$, we write

$$g_i(t) = \sum_{k=0}^{i} b_k t^k = \sum_{0 \leq j \leq i/B} G_j(t) t^{Bj}$$

where $G_j(t)$ is a polynomial of degree at most $B-1$ in t. Then

$$g_i \circ f = \sum_{0 \leq j \leq i/B} G_j(f) f^{Bj}.$$

We precompute $f_j = f^j$ for $0 \leq j \leq B$ and $F_j = f_B^j$ for $0 \leq j \leq L/B$, up to order L. Now, put $i = JB + I$ with $0 \leq I < B$. One gets

$$g_i \circ f = \sum_{0 \leq j < J} G_j(f_1) F_j + \left(\sum_{k=0}^{I-1} b_{JB+k} f_k \right) F_J + b_i f_I F_J = \Sigma_{1,i} + \Sigma_{2,i} F_J + b_i f_I F_J.$$

(We use the convention that if $I = 0$, $\Sigma_{2,i} = 0$.) It is easy to see that all terms of $\Sigma_{1,i}$ and $\Sigma_{2,i}$ up to order L (not i) depend only on the first coefficients $b_1, \ldots, b_{i-1}$. Now, it is easy to get the i-th term of $\Sigma_{2,i} F_J$ in $O(i)$ steps, as well as that of $f_I F_J$, which enables us to find the desired coefficient c_i.

Once this is done, we have to update the series. Note that we do not need the terms of indices $\leq i$. We see that if $I < B-1$, then $\Sigma_{1,i+1} = \Sigma_{1,i}$ and

$$\Sigma_{2,i+1} = \Sigma_{2,i} + b_i f_I.$$

In this case, updating the series costs $O(L-i)$. If $I = B-1$, then

$$\Sigma_{1,i+1} = \Sigma_{1,i} + (\Sigma_{2,i} + c_i f_I) F_J$$

and $\Sigma_{2,i+1} = 0$. Since we only need the terms of degree $> i$, this costs $O((L-i)^\mu)$.

Precomputing the f_j's costs $\sum_{j=2}^{B} (L-j)^\mu$, that of the F_j's is $\sum_{j=2}^{L/B} (L-jB)^\mu$ and leads to a storage of $O(B+L/B)$ series with L terms. The cost of the computations of all $\Sigma_{1,i}$ and $\Sigma_{2,i}$ is also $\sum_{j=2}^{L/B} (L-jB)^\mu$. So we need to minimize:

$$\sum_{j=2}^{B} (L-j)^\mu + 2 \sum_{j=2}^{L/B} (L-jB)^\mu.$$

After some computations, we find

Proposition 6.4. *The cost of the incremental version of Brent and Kung is minimal for $B = \sqrt{\frac{2}{\mu+1}} L^{1/2}$, giving a running time of approximately $2BL^{1/2}$ with a storage of $O(L^{1/2})$.*

7. Efficient implementation of Couveignes's algorithm

In this section, we give the algorithms needed to implement Couveignes's ideas, and deduce from this the complexity of the method. We will note $\Psi(t)$ for $\Psi_{p,E}(t)$ and $\mathcal{R}(t,s)$ for $\mathcal{R}_{p,E}(t,s)$; $\Psi^*(t)$ for $\Psi_{p,E^*}(t)$ and $\mathcal{R}^*(t,s)$ for $\mathcal{R}_{p,E^*}(t,s)$.

7.1. Precomputations for p alone.

7.1.1. *Multiplication by p.* The first thing we need is to compute the multiplication by p and the fraction $\mathcal{R}(t,s)$, as indicated in section 4.3. These computations do not depend on ℓ. The cost is $O(p^2 \log p)$ elementary operations.

7.1.2. *Solving $X - X^p = \alpha$.* We use the following result due to Hilbert (see for example [34]):

Proposition 7.1. *The equation*

$$\beta - \beta^p = \alpha \tag{19}$$

has a solution in $\mathbb{K}$ if and only if $\mathrm{Tr}_{\mathbb{K}/\mathbb{F}_p}(\alpha) = 0$. Moreover, if θ has trace 1, then a solution of this equation is:

$$\beta = \alpha\theta^p + (\alpha + \alpha^p)\theta^{p^2} + \cdots + (\alpha + \alpha^p + \cdots + \alpha^{p^{n-2}})\theta^{p^{n-1}}. \tag{20}$$

Remark that if (19) has a solution β, then $\beta + k$ is a solution for all k in the prime field $\mathbb{F}_p$. It is also easy to see that the map $\alpha \mapsto \beta$ is linear. Having computed the matrix of this application, all equations (16) can be solved by applying this matrix to the coefficients of this equation.

Note also the very important fact that the computation of this matrix depends only on p and n and not on ℓ. This means that it can be performed only once before any isogeny computation. The cost of setting up this matrix can thus be neglected. Note that we need to store $O(n^2)$ elements in $\mathbb{F}_p$ and that the time needed to apply the matrix is $O(\log p)$ (multiplications in $\mathbb{K}$).

7.2. **Finding one morphism.** We distinguish two steps: a precomputation step and then the actual computation.

7.2.1. *Precomputation phase.* Series which are independent of $\mathcal{U}$ are completely computed while only a few terms of the other series can be initialized. We also perform some precomputations for use in the composition of series as in section 6.2. We assume we want $\mathcal{L}$ terms of $\mathcal{U}$.

Precisely, we precompute:

(1) $\mathcal{S}(\tau)_{\mathcal{L}+1}$ from $(\tau)_{\mathcal{L}}$ with proposition 6.1;
(2) A such that $(1+A)^i \neq 1 + A^i$ for all $i \leq \mathcal{L}$ (this implies in particular that $q > \mathcal{L}$).
(3) $\mathcal{S}(A\tau)_{\mathcal{L}+1}$ from $\mathcal{S}(\tau)_{\mathcal{L}+1}$.
(4) The series
$$((\tau \oplus A\tau(\tau))_{\mathcal{L}}, \mathcal{S}^*(\tau \oplus A\tau(\tau))_{\mathcal{L}+1}) = ((\tau)_{\mathcal{L}}, \mathcal{S}(\tau)_{\mathcal{L}+1}) \oplus ((A\tau)_{\mathcal{L}}, \mathcal{S}(A\tau)_{\mathcal{L}+1})$$
from the addition law of section 4.2, as well as the powers needed in the composition of series.
(5) The truncated series $\Psi(\tau)_{p^r} = \mathcal{R}(\tau, \mathcal{S}(\tau))_{p^r+1}$ and its powers up to the order needed for the fast substitution algorithm. (See section 6.2.)
(6) All the intermediate series to compute $\mathcal{R}^*(\mathcal{U}(\tau), \mathcal{S}^*(\mathcal{U}(\tau)))$ as far as possible as in proposition 6.3. For instance, in characteristic 2,
$$\mathcal{R}^*(t,s) = \frac{t^3 + \tilde{a}_6^* s^2 t}{t^2 + \tilde{a}_6^* s^2 + \tilde{a}_6^* t^2 s + s + ts + t^3 + \tilde{a}_6^* t s^2}$$
and we initialize all the monomials of this fraction once substituted $(\mathcal{U}(\tau), \mathcal{S}^*(\mathcal{U}(\tau)))$ for (t,s); $\mathcal{U}(\tau)_1 = \tau$, $\mathcal{S}^*(\mathcal{U}(\tau))_3 = \tau^3$, $\mathcal{U}(\tau)\mathcal{S}^*(\mathcal{U}(\tau))_4 = \tau^4$, $\mathcal{U}^2(\tau)_2 = \tau^2$, $\mathcal{S}^*(\mathcal{U}(\tau))_6^2 = \tau^6$, ...
(7) As in step 6, all the intermediate series to compute $\mathcal{U}(\tau) \oplus \mathcal{U}(A\tau)$ as in the proof of proposition 6.2; $\mathcal{U}(A\tau)_1 = A\tau$, $\mathcal{S}^*(\mathcal{U}(\tau))_3 = A^3\tau^3$, $\lambda(\tau)_2 = (A^2 + A + 1)\tau^2$, $\nu(\tau)_3 = (A^2 + A)\tau^3$, ...

Complexity considerations: We summarize the time and space complexity in the following table.

step	1	2	3	4	5	6	7
time	$O(\mathcal{L}^2)$	$O(\mathcal{L})$	$O(\mathcal{L})$	$O(\mathcal{L}^\mu)$	$O(p\mathcal{L}^\mu)$	$O(p)$	$O(p)$
space	$O(\mathcal{L})$		$O(\mathcal{L})$	$O(\mathcal{L}^{3/2})$	$O(\mathcal{L}^{3/2})$	$O(p)$	$O(p)$

We conclude that the time complexity of this phase is at most $O(\max(p\mathcal{L}^\mu, \mathcal{L}^2))$ with a storage at most $O(\mathcal{L}^{3/2})$.

7.2.2. *Finding the morphism.* At the beginning of the i^{th} iteration, $\mathcal{U}(\tau)_{i-1}$ is known and as far as the intermediate series are concerned, $\mathcal{S}(\mathcal{U}(\tau))_{i+1}$, $\mathcal{U}(A\tau)_{i-1}$, $\mathcal{S}(\mathcal{U}(\tau))_{i+1}$, $\lambda(\tau)_i$, $\nu(\tau)_{i+1}$ ... are known by proposition 6.1. Then, formal computations enable us to compute u_i whose knowledge allows us to update the intermediate series in order to be ready for the $(i+1)^{\text{th}}$ iteration. We study the cases $i \neq p^e$ and $i = p^e$ separately.

The case $i \neq p^e$: We find u_i using $\mathcal{U}(\tau \oplus A\tau) = \mathcal{U}(\tau) \oplus \mathcal{U}(A\tau)$.

Step 1-a: We need to compute the i^{th} coefficient of $\mathcal{U}(\tau \oplus A\tau)$. We do this using the algorithm described in section 6.2. We get an equation of the type $(1+A)^i u_i + d$.

Step 1-b: We have to calculate the i^{th} coefficient of $\mathcal{U}(\tau) \oplus \mathcal{U}(A\tau)$ as a function of u_i. Since each intermediate series needed for computing this coefficient is known up to i, one can obtain as a function of u_i, the $(i+2)^{\text{th}}$ coefficient of $\mathcal{S}(\mathcal{U}(\tau))$, the i^{th} coefficient of $\mathcal{U}(A\tau)$, the $(i+2)^{\text{th}}$ coefficient of $\mathcal{S}(\mathcal{U}(\tau))$, the $(i+1)^{\text{th}}$ coefficient of $\lambda(\tau)$ and so on. Finally the coefficient we are looking for is equal to $(1+A^i)u_i + b$.

The complexity of this phase is $O(i)$.

The case $i = p^e$: We find u_i with equation $[p] \circ \mathcal{U} = \mathcal{U} \circ [p]$ that we rewrite as

$$\tilde{\mathcal{U}}(\Psi(\tau)) = \mathcal{R}^*(\mathcal{U}(\tau), \mathcal{S}^*(\mathcal{U}(\tau))).$$

This enables us to use the same techniques as the one described just above, namely applying $\tilde{\mathcal{U}}$ to a known series using precomputations and computing a rational fraction in two series.

Step 2-a: we compute as a function of u_i the i^{th} coefficient of $\tilde{\mathcal{U}}(\tau) \circ \Psi(\tau)$. This is done as in step 1-a. This coefficient is equal to $a\tilde{u}_i + d$.

Step 2-b: we compute formally the i^{th} coefficient of $\mathcal{R}^*(\mathcal{U}(\tau), \mathcal{S}^*(\mathcal{U}(\tau)))$. To perform that, we proceed as in step 1-b. We have to compute as a function of u_i the $(i+2)^{\text{th}}$ coefficient of $\mathcal{S}^*(\mathcal{U}(\tau))$, the $(i+2)^{\text{th}}$ coefficient of $\mathcal{S}^*(\mathcal{U}(\tau))$, ... This coefficient is equal to $u_i + b$.

Finally $u_i^p - a^p u_i + b^p - d^p = 0$ and we choose one of the roots for u_i.

We update the intermediate series, that is to say we obtain from $\mathcal{U}(\tau)_i$ the intermediate truncated series $\mathcal{S}(\mathcal{U}(\tau))_{i+2}$, $\mathcal{U}(A\tau)_i$, $\mathcal{S}(\mathcal{U}(\tau))_{i+2}$, $\lambda(\tau)_{i+1}$, $\nu(\tau)_{i+2}$, etc.

Complexity: We see that the computation of one morphism is dominated by the composition of series. Hence, the overall cost of this is $O(\mathcal{L}^{\mu+1/2}) = O(\ell^{\mu+1/2})$. All intermediate series will need up to $O(p\mathcal{L})$ terms.

7.3. **Isogeny testing.** Suppose we are given a morphism $\mathcal{M}(t)$ between $\mathcal{E}$ and $\mathcal{E}^*$. Put

$$Z^*(t) = \frac{\mathcal{S}^*(\mathcal{M}(t))}{\mathcal{M}(t)}$$

and we want to find a series $\hat{M}$ such that $Z^*(t) = \hat{M}(Z(t))$. Once we have done this, we need to compute a fraction whose expansion coincides with that of $\hat{M}$.

7.3.1. *From $\mathcal{M}$ to $\hat{M}$.* We know the expansions of $Z(t) = t^2 + a_1 t^3 + (a_1^2 + a_2)t^4 + O(t^5)$ and $Z^*(t) = m_2 t^2 + \cdots + m_{4\ell+1} t^{4\ell+1} + O(t^{4\ell+2})$. We are looking for the coefficients of $\hat{M}(u) = \hat{m}_1 u + \hat{m}_3 u^3 + \cdots + \hat{m}_{2\ell+1} u^{2\ell+1} + O(u^{2\ell+2})$. We will find these coefficients one at a time. Since we will have to perform many isogeny tests, it is worth precomputing all odd powers of $Z(t)$, namely $Z_i(t) = Z(t)^i$ for $1 \le i \le 4\ell + 1$, i odd. This takes $O(\ell^2)$ elements. This precomputation phase requires $O(\ell^{\mu+1})$ operations and is done only once for all ℓ (which really means we do that for the maximal value of ℓ to be used). The procedure is the following:

procedure RecoverSeriesInZ

(1) $\hat{m}_1 := m_2$; $W := W - \hat{m}_1 Z_1$;
(2) **for** $i = 1$ **to** ℓ **do**
  {at this point, $W = wt^{2i+1} + O(t^{2i+2})$}
  (a) $\hat{m}_{2i+1} := w$;
  (b) $W := W - \hat{m}_{2i+1} Z_{2i+1}$;

The computation phase takes $O(\ell^2)$ operations.

7.3.2. *Recovering the fraction.* Assume $F(z) = f_0 + f_1 z + \cdots + f_m z^m$ and $G(z) = g_0 + g_1 z + \cdots + g_m z^m$ are two polynomials of $\mathbb{K}[z]$. Then

$$\frac{F(z)}{G(z)} = A(z) = \sum_{k=0}^{\infty} a_k z^k$$

where the a_k's satisfy recurrence relations deduced from the coefficients of G.

Conversely, given a series $A(z)$, known up to order $2m$, we can compute its (m, m) Padé approximant defined as a rational fraction $U(z)/V(z)$ with $\deg(U) \le m$, $\deg(V) \le m$ and

$$A(z)V(z) - U(z) = O(z^{2m+1}).$$

The (m, m) approximant can be computed using Berlekamp's algorithm [33] in $O(m^2)$ operations or using algorithm EMGCD of [6] in $O(m(\log m)^2)$ operations. Note that from a practical point of view, Berlekamp's algorithm is faster.

7.3.3. *The final algorithm and its complexity.* The isogeny test can be summarized as follows:

procedure IsogenyTest(ℓ, $\mathcal{M}(t)$, $\mathcal{S}^*(\mathcal{M}(t))$)

(1) compute $Z^*(t) = \mathcal{S}^*(\mathcal{M}(t))/\mathcal{M}(t)$;
(2) compute $\hat{M}(Z) = \hat{m}_1 Z + \hat{m}_3 Z^3 + \cdots + \hat{m}_{2\ell+1} Z^{2\ell+1} + O(Z^{2\ell+2})$ using algorithm RecoverSeriesInZ;
(3) recover the fraction $F(Z)/G(Z)$ which is a $(\ell+1, \ell+1)$ Padé approximant of $\hat{M}(Z)$; at this point, F and G have degree $\leq \ell+1$ *a priori*;
(4) if $\deg(F) = \deg(G) = \ell$ and F is Z times the square of a polynomial, then $\hat{M}$ comes from the isogeny we are looking for.

The first step takes $O(\ell^\mu)$ operations, the second $O(\ell^2)$ which dominates the third step. Therefore, we see that the cost of the isogeny test is $O(\ell^2)$.

Note also that in the "multiplication" strategy, one already has $\mathcal{S}^*(\mathcal{M}(t))$ at one's disposal.

7.4. **Enumerating all the morphisms.**

7.4.1. *Backtracking.* It is easy to see that the cost of this approach is $O(\mathcal{L})$ times the cost of finding one morphism plus that of an isogeny test. The total cost is thus $O(\ell^{\max(\mu+3/2,3)})$.

7.4.2. *Multiplication by a p-adic integer.* In fact, we do not really multiply by a p-adic integer, but merely perform additions in the formal group, until we find the isogeny. The algorithm is as follows:

procedure ComputeIsogeny(ℓ, $\mathcal{E}$, $\mathcal{E}^*$)

(1) compute a generator $\mathcal{U}$ of the set of morphisms between $\mathcal{E}$ and $\mathcal{E}^*$ using the algorithms of section 7.2;
(2) **for** $N = 1$ to $p^{r+1}/2$ and N prime to p **do**
   (a) compute $(\mathcal{M}(t), \mathcal{S}^*(\mathcal{M}(t))) = [N] \circ (\mathcal{U}(t), \mathcal{S}(\mathcal{U}(t)))$;
   (b) use IsogenyTest to test whether $\mathcal{M}$ comes from an isogeny; if yes, stop.

Note that we compute $(\mathcal{M}(t), \mathcal{S}^*(\mathcal{M}(t)))$ using a formal addition between the preceding computed value and $(\mathcal{U}(t), \mathcal{S}(\mathcal{U}(t)))$ or $[2](\mathcal{U}(t), \mathcal{S}(\mathcal{U}(t)))$.

The cost of the second approach is the cost of finding one morphism, $O(\mathcal{L}^{\mu+1/2})$ multiplications, plus $O(\mathcal{L})$ times the cost of an addition in the formal group – $O(\mathcal{L}^\mu)$ multiplications – plus $O(\mathcal{L})$ times the cost of the isogeny test of cost $O(\mathcal{L}^2)$. So, the complexity of this second approach is $O(\ell^{\max(\mu+1,3)})$.

7.4.3. *Complexity and choice of the method.* Asymptotically, if $\mu \leq 3/2$, both methods have the same complexity $O(\ell^3)$. If $\mu > 3/2$, the second one is better and the complexity is $O(\ell^3)$. However, from a practical point of view, the second approach is always better, since, apart from the isogeny test, we replace a substitution of series whose complexity is $O(\mathcal{L}^{\mu+1/2})$ by a formal addition whose complexity is only $O(\mathcal{L}^\mu)$ (remember that $1 \leq \mu \leq 2$).

7.5. **Overall complexity.** We summarize the preceding results.

Proposition 7.2. *After preprocessing, the cost of Couveignes's algorithm is $O(\ell^3)$. The storage is $O(\ell^2)$.*

8. Implementation in characteristic 2

We give an example of our implementation of Couveignes's idea for $q = 2^n$. Let $E : Y^2 + XY = X^3 + a_6$ be an elliptic curve.

8.1. **Simplifying formulas.** After the classical change of variables $t = -x/y$, $s = -1/y$ to set $O_E = (0,0)$, the equation of $\mathcal{E}$ becomes

$$s = t^3 + ts + a_6 s^3. \tag{21}$$

In the special case $t(\tau) = \tau$, the coefficients of the series $\mathcal{S}(\tau) = \sum_{i=3}^{\infty} \mathcal{S}_i \tau^i$ are as follows:

i	3	4	5	6	7	8	9	10	11
$\mathcal{S}_i$	1	1	1	1	1	1	$1 + a_6$	1	1

and for $i \geq 6$:

$$\mathcal{S}_{2i} = \mathcal{S}_{2i-1} + a_6 \left(\mathcal{S}_{2i-6} + \sum_{j=4}^{i-2} \mathcal{S}_j^2 \mathcal{S}_{2i-2j} \right), \mathcal{S}_{2i+1} = \mathcal{S}_{2i} + a_6 \left(\mathcal{S}_{2i-5} + \mathcal{S}_{i-1}^2 + \sum_{j=4}^{i-2} \mathcal{S}_j^2 \mathcal{S}_{2i-2j+1} \right).$$

Using standard tools [43], we also find that $\mathcal{S}(t)$ satisfies the differential equation:

$$\left(-54\,a_6\,t^5 - 4\,t^3 + 14\,t^2 - 18\,t + 8\right) y + \left(54\,t^6 a_6 + 6\,t^4 - 28\,t^3 + 54\,t^2 - 48\,t + 16\right) y' + \left(-27\,t^8 a_6 + 54\,t^7 a6 - 4\,t^5 + 20\,t^4 - 36\,t^3 + 28\,t^2 - 8\,t\right) y'' = 0$$

over $\mathbb{Z}[a_6]$, from which we find that the s_i's satisfy the recurrence relation

$$27a_6 n\,(-n+1)\,s(n) + 54\,a_6\,n\,(2+n)\,s(n+1) - 2\left(2\,n^2 + 7\,n + 5\right) s(n+3) + 2\left(10\,n^2 + 56\,n + 71\right) s(n+4)$$
$$-18\left(2\,n^2 + 15\,n + 26\right) s(n+5) + 4\left(7\,n^2 + 65\,n + 140\right) s(n+6) - 8\left(n^2 + 11\,n + 28\right) s(n+7) = 0$$

together with the initial values: $s_0 = s_1 = s_2 = 0, s_3 = s_4 = s_5 = s_6 = 1$. Using these formulas, we can compute the s_n over $\mathbb{Z}[a_6]$ and then reduce them modulo 2.

We can rewrite the formulae of section 4.2 in order to decrease their computational cost. In particular, if $t_1 \neq t_2$, $t_1 \oplus t_2$ is computed as:

$$t_1 \oplus t_2(\tau) = \frac{t_1(\tau) + t_2(\tau) + \lambda(\tau) + a_6\lambda^2(\tau)(s_1(\tau) + s_2(\tau) + \nu(\tau))}{1 + t_1(\tau) + t_2(\tau) + \lambda(\tau) + a_6\lambda^2(\tau)\,(s_1(\tau) + s_2(\tau) + \nu(\tau) + \lambda(\tau))}.$$

Arranging computations so as to reuse series already computed, adding two distinct formal points requires 4 multiplications of series and 2 divisions. For computing $([2]t(\tau), \mathcal{S}([2]t(\tau))) = [2](t(\tau), s(\tau))$, we use (11). This computation costs 4 multiplications of series and 2 divisions.

8.2. **Example.** Let $\mathbb{K} = \mathbb{F}_{2^8} = \mathbb{F}_2[T]/(T^8 + T^4 + T^3 + T + 1)$. Every element of $\mathbb{K}$ can be written as a polynomial in T. In order to reduce the space needed to write the different results, we will write such a polynomial $a(T) = \sum_{i=0}^{n-1} a_i T^i$ as $\overline{a(2)}$. For instance, the polynomial $T^2 + T$ will be abbreviated as $\overline{6}$.

Let us compute an isogeny of degree $\ell = 5$ between $E = [\overline{7}]$ and $E^* = [\overline{8}]$. We first find that $A = \overline{2}$ is valid. The equation for u_2, coming from equating $[2] \circ \mathcal{U} = \mathcal{U} \circ [2]$ is

$$\sqrt{u_2}^2 + \sqrt{u_2} = 0$$

and we select $u_2 = 0$. Next, we find that u_3 is a root of

$$\overline{6}\,u_2 + \overline{6}\,u_3 = 0$$

which gives $u_3 = 0$. For u_4, we have:

$$\sqrt{u_4}^2 + \sqrt{u_4} + \sqrt{\overline{15}} = 0$$

and we choose $u_4 = \overline{56}$. Once all computations are done, we find

$$\begin{aligned}\mathcal{U}(t) \;=\;& t + \overline{56}t^4 + \overline{56}t^5 + \overline{15}t^7 + \overline{16}t^8 + \overline{31}t^9 + \overline{219}t^{10} + \overline{124}t^{11} + \overline{5}t^{12} + \overline{44}t^{13} + \overline{91}t^{14} \\ &+\overline{47}t^{15} + \overline{210}t^{16} + \overline{201}t^{17} + \overline{231}t^{18} + \overline{198}t^{19} + \overline{188}t^{20} + \overline{118}t^{21} + O(t^{22}).\end{aligned}$$

We first have that

$$\begin{aligned}Z(t) = \frac{\mathcal{S}(t)}{t} \;=\;& t^2 + t^3 + t^4 + t^5 + t^6 + t^7 + \overline{6}t^8 + t^9 + t^{10} + t^{11} + \overline{6}t^{12} + t^{13} + \overline{20}t^{14} \\ &+\overline{20}t^{15} + \overline{6}t^{16} + t^{17} + t^{18} + t^{19} + \overline{6}t^{20} + t^{21} + \overline{20}t^{22} + O(t^{23}).\end{aligned}$$

Now we have to look for N, $1 \leq N \leq 16$, N odd, such that $[N] \circ \mathcal{U}$ is the series associated with $\mathcal{I}$. After a first test, it turns out that $\mathcal{U}$ is not the morphism we are looking for. On the other hand, we have

$$\begin{aligned}[3](\mathcal{U}(t)) \;=\;& t + t^2 + t^3 + \overline{56}t^4 + \overline{56}t^5 + \overline{56}t^6 + \overline{55}t^7 + \overline{39}t^8 + \overline{39}t^9 + \overline{244}t^{10} + \overline{84}t^{11} + \overline{154}t^{12} \\ &+\overline{28}t^{13} + \overline{79}t^{14} + \overline{52}t^{15} + \overline{247}t^{16} + \overline{51}t^{17} + \overline{44}t^{18} + \overline{66}t^{19} + \overline{102}t^{20} + \overline{84}t^{21} + O(t^{22})\end{aligned}$$

from which

$$\begin{aligned}Z^*(t) = \frac{\mathcal{S}^*(U(t))}{U(t)} \;=\;& t^2 + t^3 + t^4 + t^5 + \overline{56}t^6 + \overline{56}t^7 + \overline{6}t^8 + t^9 + \overline{39}t^{10} + \overline{39}t^{11} + \\ &\overline{182}t^{12} + \overline{30}t^{13} + \overline{143}t^{14} + \overline{32}t^{15} + \overline{91}t^{16} + t^{17} + \overline{241}t^{18} + \\ &\overline{241}t^{19} + \overline{67}t^{20} + \overline{200}t^{21} + \overline{138}t^{22} + O(t^{23}),\end{aligned}$$

which can be rewritten as

$$Z(t) + \overline{57}Z^3(t) + \overline{31}Z^5(t) + \overline{13}Z^7(t) + \overline{214}Z^9(t) + \overline{120}Z^{11}(t) + O(t^{23}).$$

Now we use the Berlekamp Massey algorithm to recover the fraction, which in this particular case gives

$$\frac{\overline{140}Z^5(t) + \overline{15}Z^3(t) + Z(t)}{\overline{239}Z^4(t) + \overline{54}Z^2(t) + 1},$$

n	$f(T)$	c
155	$T^7+T^5+T^4+1$	80860670297104421704641
196	T^3+1	168959031790830995673970347393
300	T^5+1	−1057128182990122062466877450474871210809126 3
400	$T^5+T^3+T^2+1$	831311719593373938759692923178171920626211278774 17\ 9820465793
500	$T^8+T^6+T^5+T^2+T+1$	−1022525379417220053537215648371674704330886180912\ 846154234245338259365269 75
601	$T^7+T^4+T^3+T^2+T+1$	377757427631721806546376981799227629798971729208 00\ 677014581466240683645488986670133490096 65
701	$T^9+T^8+T^7+T^4+T^2+1$	−635995503420894831900031121630947891732157980332 9\ 465179598270185421050043964651481876644528892263 58\ 9295359
1009	$T^{11}+T^4+T^2+1$	550079058499346141446244095017123794191976346205 24\ 534567632260483655377597058213876976282320229650 34\ 095450594133404979993418055065277722637699785638630 5

TABLE 1. Values of c such that $\#E_X = q+1-c$.

	$\mathbb{F}_{2^{155}}$	$\mathbb{F}_{2^{196}}$	$\mathbb{F}_{2^{300}}$	$\mathbb{F}_{2^{701}}$	$\mathbb{F}_{2^{1009}}$
$\ell_{\max}$	59	73	109	337	577
Isogenies	24	580	22974	4842770	13457961
Total	217	1381	30221	7897343	21018853

TABLE 2. Records for the second implementation

and via $Z(t)=1/X(t)$, we obtain

$$\mathcal{I}(X(t)) = \frac{X^5(t)+\overline{15}X^3(t)+\overline{140}X(t)}{(X^2(t)+\overline{57}X(t)+\overline{74})^2}.$$

8.3. **Implementation and results.**

8.3.1. *Benchmarks.* In [30], we benchmarked our implementation using curves defined over small finite fields, as was done in [22]. We also explained in the same paper how we can tune our program using parameters describing several strategies. In the case of the field $\mathbb{K}=\mathbb{F}_{2^{300}}$, we have made precise statistics on every part of the algorithm. The results are given in the tables given at the end of the article. In Table 3, one finds for each Elkies prime power ℓ^d the time `Prec` needed for the precomputations, $\oplus$ designates the time of a formal addition, `BM` the time for the Berlekamp-Massey algorithm and N is the 2-adic integer such that $[N]\circ\mathcal{U}$ comes from an isogeny (we take all coefficients $u_{2^i}=P_i(T)$ of $\mathcal{U}$ such that $P_i(0)=0$).

8.3.2. *Records.* In [37, 36], the authors gave timings for larger fields $\mathbb{F}_{2^{155}}$ and $\mathbb{F}_{2^{195}}$. For these fields and for larger fields (the last one being the current record, $\mathbb{F}_{2^{1009}}$, as of January 1996), our implementation gave the following timings, for the curve:

$$E_X : y^2+xy = x^3+T^{16}+T^{14}+T^{13}+T^9+T^8+T^7+T^6+T^5+T^4+T^3$$

(the coefficient was chosen as the binary expression of 91128 – our zip code – converted to a polynomial if $\mathbb{F}_{2^n}$). Table 1 gives for some values of n a polynomial $f(T)$ such that $T^n+f(T)$ is a defining polynomial of $\mathbb{F}_{2^n}=\mathbb{F}_2[T]/(T^n+f(T))$ and the value of c such that $\#E_X(\mathbb{F}_{2^n})=2^n+1-c$.

The interested reader can find the timings for our first implementation in [30], as well as a comparison with the case of prime fields of large characteristic. Table 2 refers to the implementation that uses all features described in the present article. All these records have been done using a network of DecAlpha workstations, using an obvious distributed implementation of the algorithm.

9. Conclusion

There are basically two approaches for computing isogenies between elliptic curves over a finite field. The Atkin-Elkies method works well when the characteristic is large and Couveignes's method for the small characteristic. In the particular case of the characteristic 2, a new method has being developed by the first author [28]. This method does not use formal groups and is much faster than Couveignes's in practice.

In the general case of p small, Couveignes [10] (and [11] for a more detailed version) has very recently proposed a new algorithm that uses properties of the p-torsion to compute the isogenies. The implementation and comparison of these new methods are currently being done by the first author (see [32] for a comparison of the algorithms and [29] for more details).

Acknowledgments. First of all, the authors want to express their gratitude to J.-M. Couveignes, without whom this work could not have been possible. Many thanks to J.-M. Steyaert for his careful reading and for pointing out to us the use of GFUN and the theory lying behind it.

References

[1] Atkin, A. O. L. The number of points on an elliptic curve modulo a prime. Draft, 1988.
[2] Atkin, A. O. L. The number of points on an elliptic curve modulo a prime (ii). Draft. Available on `http://listserv.nodak.edu/archives/nmbrthry.html`, 1992.
[3] Atkin, A. O. L., and Morain, F. Elliptic curves and primality proving. *Math. Comp. 61*, 203 (July 1993), 29–68.
[4] Borel, A., Chowla, S., Herz, C. S., Iwasawa, K., and Serre, J.-P. *Seminar on complex multiplication.* No. 21 in Lecture Notes in Math. Springer, 1966.
[5] Bosma, W. Primality testing using elliptic curves. Tech. Rep. 85-12, Math. Instituut, Universiteit van Amsterdam, 1985.
[6] Brent, R. P., Gustavson, F. G., and Yun, D. Y. Y. Fast solution of Toeplitz systems of equations and computation of Padé approximants. *Journal of Algorithms 1* (1980), 259–295.
[7] Brent, R. P., and Kung, H. T. Fast algorithms for manipulating formal power series. *Journal of the ACM 25* (1978), 581–595.
[8] Comtet, L. Calcul pratique des coefficients de Taylor d'une fonction algébrique. *Enseignement Math. 10* (1964), 267–270.
[9] Couveignes, J.-M. *Quelques calculs en théorie des nombres.* Thèse, Université de Bordeaux I, July 1994.
[10] Couveignes, J.-M. Computing l-isogenies using the p-torsion. In *ANTS-II* (1996), H. Cohen, Ed., vol. 1122 of *Lecture Notes in Comput. Sci.*, Springer-Verlag, pp. 59–65.
[11] Couveignes, J. M. Isomorphisms between Artin-Schreier towers. Draft, available on `http://www.math.u-bordeaux.fr/~couveign`, Jan. 1997.
[12] Couveignes, J.-M., Dewaghe, L., and Morain, F. Isogeny cycles and the Schoof-Elkies-Atkin algorithm. Research Report LIX/RR/96/03, LIX, Apr. 1996.
[13] Couveignes, J.-M., and Morain, F. Schoof's algorithm and isogeny cycles. In *ANTS-I* (May 1994), L. Adleman and M. D. Huangs, Eds., vol. 877 of *Lecture Notes in Comput. Sci.*, Springer-Verlag, pp. 43–58.
[14] Deuring, M. Die Typen der Multiplikatorenringe elliptischer Funktionenkörper. *Abh. Math. Sem. Hamburg 14* (1941), 197–272.
[15] Dewaghe, L. Un corollaire aux formules de Vélu. Preprint, Dec. 1995.
[16] Elkies, N. D. Explicit isogenies. Draft, 1991.
[17] Elkies, N. D. Elliptic and modular curves over finite fields and related computational issues. In *Computational perspectives on number theory (Chicago, IL, 1995)*, vol. 7 of *AMS/IP Stud. Adv. Math.* Amer. Math. Soc., Providence, RI, 1998, pp. 21–76.
[18] Fricke, R. *Die elliptischen Funktionen und ihre Anwendungen.* Teubner, Leipzig, 1992.
[19] Fröhlich, A. *Formal groups*, vol. 74 of *Lecture Notes in Math.* Springer-Verlag, 1968.
[20] Goldwasser, S., and Kilian, J. Almost all primes can be quickly certified. In *Proc. 18th STOC* (1986), ACM, pp. 316–329. May 28–30, Berkeley.
[21] Gunji, H. The Hasse invariant and p-division points of an elliptic curve. *Arch. Math. 27*, 2 (1976), 148–158.
[22] Harper, G., Menezes, A., and Vanstone, S. Public-key cryptosystems with very small key length. In *Advances in Cryptoloy – EUROCRYPT '92* (1993), R. A. Rueppel, Ed., vol. 658 of *Lecture Notes in Comput. Sci.*, Springer-Verlag, pp. 163–173. Workshop on the Theory and Application of Cryptographic Techniques, Balatonfüred, Hungary, May 24–28, 1992, Proceedings.
[23] Husemöller, D. *Elliptic curves*, vol. 111 of *Graduate Texts in Mathematics.* Springer, 1987.
[24] Kaneko, M., and Zagier, D. Supersingular j-invariants, hypergeometric series, and Atkin's orthogonal polynomials. In *Computational perspectives on number theory (Chicago, IL, 1995)*, vol. 7 of *AMS/IP Stud. Adv. Math.* Amer. Math. Soc., Providence, RI, 1998, pp. 97–126.
[25] Koblitz, N. Elliptic curve cryptosystems. *Math. Comp. 48*, 177 (Jan. 1987), 203–209.
[26] Lehmann, F., Maurer, M., Müller, V., and Shoup, V. Counting the number of points on elliptic curves over finite fields of characteristic greater than three. In *ANTS-I* (1994), L. Adleman and M.-D. Huang, Eds., vol. 877 of *Lecture*

Notes in Comput. Sci., Springer-Verlag, pp. 60–70. 1st Algorithmic Number Theory Symposium - Cornell University, May 6-9, 1994.
[27] Lenstra, Jr., H. W. Factoring integers with elliptic curves. *Annals of Math. (2) 126* (1987), 649–673.
[28] Lercier, R. Computing isogenies in $\mathbf{F}_{2^n}$. In *Algorithmic number theory (Talence, 1996)*, vol. 1122 of *Lecture Notes in Comput. Sci.* Springer, Berlin, 1996, pp. 197–212.
[29] Lercier, R. *Algorithmique des courbes elliptiques dans les corps finis*. Thèse, École polytechnique, June 1997.
[30] Lercier, R., and Morain, F. Counting the number of points on elliptic curves over finite fields: strategies and performances. In *Advances in Cryptology – EUROCRYPT '95* (1995), L. C. Guillou and J.-J. Quisquater, Eds., no. 921 in Lecture Notes in Comput. Sci., pp. 79–94. International Conference on the Theory and Application of Cryptographic Techniques, Saint-Malo, France, May 1995, Proceedings.
[31] Lercier, R., and Morain, F. Counting the number of points on elliptic curves over F_{p^n} using Couveignes's algorithm. Submitted at Math. Comp., 1995.
[32] Lercier, R., and Morain, F. Algorithms for computing isogenies between elliptic curves. In *Computational perspectives on number theory (Chicago, IL, 1995)*, vol. 7 of *AMS/IP Stud. Adv. Math.* Amer. Math. Soc., Providence, RI, 1998, pp. 77–96.
[33] Massey, J. L. Shift-register and BCH decoding. *IEEE Trans. on Information Theory IT-15*, 1 (Jan. 1969), 122–127.
[34] McEliece, R. *Finite fields for computer scientists and engineers*. Kluwer international series in engineering and computer science. Kluwer Academic Publishers, 1988.
[35] Menezes, A., and Vanstone, S. Isomorphism classes of elliptic curves over finite fields of characteristic 2. *Utilitas Math. 38* (1990), 135–153.
[36] Menezes, A. J. *Elliptic curve public key cryptosystems*. Kluwer Academic Publishers, 1993.
[37] Menezes, A. J., Vanstone, S. A., and Zuccherato, R. J. Counting points on elliptic curves over F_{2^m}. *Math. Comp. 60*, 201 (Jan. 1993), 407–420.
[38] Miller, V. S. Use of elliptic curves in cryptography. In *Advances in Cryptology - Crypto '85* (Berlin, 1986), H. C. Williams, Ed., Springer-Verlag, pp. 417–428. Lecture Notes in Computer Science Volume 218.
[39] Montgomery, P. L. Speeding the Pollard and elliptic curve methods of factorization. *Math. Comp. 48*, 177 (Jan. 1987), 243–264.
[40] Morain, F. Calcul du nombre de points sur une courbe elliptique dans un corps fini : aspects algorithmiques. *J. Théor. Nombres Bordeaux 7* (Feb. 1995), 255–282.
[41] Morain, F. Classes d'isomorphismes des courbes elliptiques supersingulières en caractéristique ≥ 3. *Util. Math. 52* (1997), 241–253.
[42] Müller, V. *Ein Algorithmus zur bestimmung der Punktanzahl elliptisher kurven über endlichen körpen der charakteristik größer drei*. PhD thesis, Technischen Fakultät der Universität des Saarlandes, February 1995.
[43] Salvy, B., and Zimmermann, P. GFUN, a MAPLE package for the manipulation of generating and holonomic functions in one variable. *ACM Transactions on Mathematical Software 20*, 2 (1994), 163–177.
[44] Schoof, R. Counting points on elliptic curves over finite fields. *J. Théor. Nombres Bordeaux 7* (1995), 219–254. Available at `http://www.emath.fr/Maths/Jtnb/jtnb1995-1.html`.
[45] Silverman, J. H. *The arithmetic of elliptic curves*, vol. 106 of *Graduate Texts in Mathematics*. Springer, 1986.
[46] Tate, J. Endomorphisms of Abelian varieties over finite fields. *Inventiones Mathematicae 2* (1966), 134–144.

(Reynald Lercier) CELAR/SSIG, Route de Laillé, F-35170 Bruz, FRANCE
E-mail address, Reynald Lercier: `lercier@celar.fr`

(François Morain) Laboratoire d'Informatique de l'École polytechnique (LIX – UMR 7650), F-91128 Palaiseau Cedex, France
E-mail address, François Morain: `morain@lix.polytechnique.fr`

ℓ	d	Prec	$\mathcal{U}$	$Z(t)^i$	$\oplus$	$Z^* = \hat{M}(Z)$	BM	N	Isogenies	%	Tot
3	1	0	0.2	0	0	0	0	3	0.4	67	0.6
5	1	0	0.5	0	0.1	0	0	7	1.3	62	2.1
7	1	0.1	0.8	0	0.1	0	0	7	2.1	64	3.3
11	1	0.4	1.9	0.1	0.4	0	0	7	6	50	12.1
	2	0.5	2.1	0.1	0.4	0	0	7	10.4	5	220.1
19	1	5.2	5.6	0.6	1.2	0	0.1	47	56	77	72.8
31	1	2.8	13.1	2.3	2.3	0.1	0.2	63	131.4	74	177.1
37	1	7.3	21	3.7	4.5	0.2	0.3	127	446.3	86	516.3
41	1	8	25.2	5	5.2	0.2	0.4	81	343.6	78	439.6
47	1	8.9	32.3	7.5	5.8	0.3	0.5	111	525.2	84	627
59	1	11.1	49.6	13.6	7.3	0.6	0.7	103	639.2	80	798.4
61	1	11.3	55	14.7	7.8	0.6	0.8	3	112.3	40	278.4
67	1	47.6	68.9	18.5	11.4	0.7	0.8	93	927.6	80	1157.2
79	1	28.2	94.4	29.8	15.8	1.1	1.2	133	1692.9	86	1970
97	1	33.6	136.3	52.6	20.2	1.7	1.7	79	1408.4	78	1802.3
101	1	34.6	142.5	58.6	20.5	1.8	1.8	161	2695.9	83	3259.2
103	1	35.1	147.4	62	20.7	1.8	1.9	37	853.4	70	1212.6
107	1	144.4	159.1	68.3	21.6	2	2	251	4433.4	90	4907.6
Tot		379.1	955.9	68.3	-	-	-	-	14285.8	82	17456.7

TABLE 3. Data for $\mathbb{F}_{2^{300}}$.