The extended Euclidian algorithm on polynomials, and the computational efficiency of hyperelliptic cryptosystems

Andreas Enge
Lehrstuhl für Diskrete Mathematik,
Optimierung und Operations Research
Universität Augsburg
86135 Augsburg
Deutschland
enge@math.uni-augsburg.de

November 22, 1999

Abstract

After generalising two reduction algorithms to characteristic 2, we analyse the average complexity of the arithmetic in hyperelliptic Jacobians over any finite field. To this purpose we determine the exact average number of field operations for computing the greatest common divisor of polynomials over a finite field by the extended Euclidian algorithm.

1 Hyperelliptic Cryptosystems

A wide variety of public key cryptosystems is based on the intractability of the discrete logarithm problem in certain groups. Historically the first type of group considered was the multiplicative group of a finite prime field, or more generally of any finite field. However, the index calculus method provides a subexponential attack on such cryptosystems (see, for instance, [McC90]), so that long keys are needed to achieve an acceptable level of security.

Elliptic curves over finite fields circumvent this problem, since no general subexponential algorithms are known for them, except for some easily avoided special cases ([MOV93], [FR94]; [SA98], [Sem98], [Sma99]). Hence only exponential attacks like the baby-step giant-step method or Pollard's ρ-method can be applied, and these are no longer practical if the number of points on the curve is large, say about 10^{50}. There is a broad range of literature on elliptic curves; the standard reference is [Sil86], for an elementary introduction see [Eng99b].

More recently, Jacobians of hyperelliptic curves over finite fields have been suggested for use in public key cryptosytems ([Kob89]). We give a brief overview to the extent that is needed for our paper in the next few sections. Section 1.1 introduces hyperelliptic curves and their Jacobians, while Sections 1.2 and 1.3 deal with algorithms for computing in these Jacobians. We generalise some algorithms which have been previously formulated for curves over fields of odd characteristic only to the characteristic 2 case. Finally we describe the cryptographic setting which underlies our comparative study in Section 1.4.

1.1 Hyperelliptic curves and their Jacobians

We introduce some basic facts about hyperelliptic curves; proofs can be found in the excellent elementary introduction by Menezes, Wu and Zuccherato [MWZ98].

Let $K = \mathbb{F}_q = \mathbb{F}_{p^m}$ be the finite field of characteristic p with q elements and $\overline{K}$ its algebraic closure. A *hyperelliptic curve* of *genus* g is the projective closure of a non-singular affine curve of the type

$$H : Y^2 + hY = f,$$

where $h \in K[X]$ is of degree at most g and $f \in K[X]$ is monic of degree $2g+1$. The case $g = 1$ is that of an elliptic curve, and we are concerned with curves of genus at least 2 in this article. If the characteristic of K is odd, we can apply an affine change of variables to obtain $h = 0$. We first concentrate on $\overline{K}$; then H consists of the *finite points* $P = (x, y) \in \overline{K} \times \overline{K}$ whose coordinates satisfy the equation, and an additional *point at infinity*, denoted by $\mathcal{O}$. These are in bijection with the *primes* or *valuations* of the function field $\overline{K}(H) := \overline{K}(X)[Y]/(H)$, which is a quadratic extension of the rational function field $\overline{K}(X)$. The *group of divisors* $\overline{\mathrm{Div}}(H)$ associated to H is the free abelian group over the points; the *degree* of a divisor $D = \sum_{P \in H} m_P P$ is the sum of its coefficients, $\deg D = \sum_{P \in H} m_P$. The *degree zero part* of the divisor group consists of all divisors of degree zero and is denoted by $\overline{\mathrm{Div}^0}(H)$. To a rational function in $\overline{K}(H)$ can be associated the divisor of its zeros and poles, each with the corresponding multiplicities; such a divisor is called *principal*, and all principal divisors form the subgroup $\overline{\mathrm{Prin}}(H) \subseteq \overline{\mathrm{Div}^0}(H)$. Now the *Jacobian* of H is defined as the abelian group $\overline{J}(H) := \overline{\mathrm{Div}^0}(H)/\overline{\mathrm{Prin}}(H)$; its elements are called *divisor classes*.

The conjugation on $\overline{K}(H)/\overline{K}(X)$, sending Y to $\overline{Y} = -Y - h$, acts in an obvious way on points and divisors by $P = (x, y) \mapsto \overline{P} = (x, -y - h(x))$, $\mathcal{O} \mapsto \mathcal{O}$ and $D = \sum_{P \in H} m_P P \mapsto \overline{D} = \sum_{P \in H} m_P \overline{P}$. We call P *splitting* if $P \neq \overline{P}$ and *ramified* otherwise; hence $\mathcal{O}$ is ramified.

A degree zero divisor $D = \sum_{P \in H} m_P P$ is called *semireduced* if all $m_P \geq 0$ for $P \neq \mathcal{O}$, at most one of m_P and $m_{\overline{P}}$ is positive for a splitting prime P and $m_P \in \{0, 1\}$ for a ramified prime $P \neq \mathcal{O}$. If furthermore $\sum_{P \neq \mathcal{O}} m_P \leq g$, then P is called *reduced*. Let the greatest common divisor of two degree zero divisors be defined by

$$\begin{aligned} &\gcd\left(\sum_{P \in H \setminus \{\mathcal{O}\}} m_P P - m_{\mathcal{O}} \mathcal{O}, \sum_{P \in H \setminus \{\mathcal{O}\}} n_P P - n_{\mathcal{O}} \mathcal{O}\right) \\ &:= \sum_{P \in H \setminus \{\mathcal{O}\}} \min(m_p, n_P) P - \left(\sum_{P \in H \setminus \{\mathcal{O}\}} \min(m_p, n_P)\right) \mathcal{O}. \end{aligned}$$

Any semireduced divisor can be uniquely written as $\mathrm{div}(a, b) := \gcd(\mathrm{div}(a), \mathrm{div}(Y - b))$ for a, $b \in \overline{K}[X]$, a monic, $\deg b < \deg a$ and $a | b^2 + hb - f$; it is reduced if and only if $\deg a \leq g$. Any divisor class contains a unique reduced divisor.

In order to compute in Jacobians, we must reduce all notions above to the finite field K itself. Note that the Galois group of $\overline{K}/K$ is topologically generated by the Frobenius automorphism

$$\varphi : \overline{K} \to \overline{K},\ x \mapsto x^q.$$

The Frobenius automorphism acts in the obvious manner on points and divisors by $(x, y) \mapsto (x^q, y^q)$. We call a divisor *rational* over K if it consists of complete orbits under φ, each of these orbits representing a prime of the function field $K(H)$. We define the degree of a rational prime divisor by the cardinality of the orbit. Adopting the projective point of view it is easy to see that $\mathcal{O}$ is rational of degree 1. Then we can define $\mathrm{Div}(H)$ as the free abelian group over the rational primes, and the definitions of $\mathrm{Div}^0(H)$, $\mathrm{Prin}(H)$ and the Jacobian $J(H)$ carry over from the algebraically closed case.

It can be easily seen that a reduced divisor $\mathrm{div}(a, b)$ is rational if and only if $a, b \in K[X]$; so as before, each divisor class of $J(H)$ has a unique representative $\mathrm{div}(a, b)$ with $a, b \in K[X]$, a monic, $\deg b < \deg a \leq g$ and $a | b^2 + hb - f$. Since multiplying a by a non-zero constant does not change the represented divisor, we drop the assumption that a is monic. This saves a normalisation step and yields a faster arithmetic at the expense of making the comparison of two divisors slower.

1.2 Adding divisors — Composition step

Let there be given two reduced divisors $D_1 = \mathrm{div}(a_1, b_1)$ and $D_2 = \mathrm{div}(a_2, b_2)$. The computation of a reduced representative for the class of $D_1 + D_2$ proceeds in two steps. First, a semireduced divisor $\mathrm{div}(a, b)$ in the class of $D_1 + D_2$ is determined; second, this divisor is reduced. This combination of a composition and a reduction step, inspired by analogous algorithms for binary quadratic forms and introduced in [Can87], is generally known as "Cantor's algorithm". In this section, we treat the composition algorithm.

Compute the greatest common divisor

$$d = \gcd(a_1, a_2, b_1 + b_2 + h) = u_1 a_1 + u_2 a_2 + u_3(b_1 + b_2 + h)$$

together with a representation by the extended Euclidian algorithm. Set

$$a = \frac{a_1 a_2}{d^2}$$

and

$$b = b_1 + \frac{u_1 a_1 (b_2 - b_1) + u_3 (f - b_1^2 - b_1 h)}{d} \bmod a.$$

This composition algorithm goes back to Gauß, who described an analogous procedure for composing binary quadratic forms in [Gau01], Article 242; its efficient application is due to Shanks ([Sha71]). Note that the formula for b in Algorithm 1 of [MWZ98] is less efficient than the one presented here, which is the generalisation of Formula (C_{3a}) in [Can87] to the arbitrary characteristic case.

The extended Euclidian algorithm

To complete the description of the composition step and to fix the notation for the rest of this article, we recall the extended Euclidean algorithm on two polynomials r_0 and r_1 with $\deg r_1 \leq \deg r_0$:

1. Set $u_0 = 1$, $u_1 = 0$, $v_0 = 0$ and $v_1 = 1$.

2. Compute $r_{k-1} = q_k r_k + r_{k+1}$ with $\deg r_{k+1} < \deg r_k$ for $k = 1, \ldots, t$, $u_{k+1} = u_{k-1} - q_k u_k$ and $v_{k+1} = v_{k-1} - q_k v_k$ for $k = 1, \ldots, t-1$, where t is the smallest index such that $r_{t+1} = 0$.

3. Set $u = u_t$ and $v = v_t$. Then $\gcd(r_0, r_1) = r_t = u r_0 + v r_1$.

Notice that during the loop in Step 2. we always have $r_k = u_k r_0 + v_k r_1$ for $k = 0, \ldots, t$. It follows easily from the description of the algorithm that

$$\begin{aligned} \deg q_k &= \deg r_{k-1} - \deg r_k \quad \text{for } k \geq 1, & (1)\\ \deg u_k &= \deg r_1 - \deg r_{k-1} \quad \text{for } k \geq 2, & (2)\\ \deg v_k &= \deg r_0 - \deg r_{k-1} \quad \text{for } k \geq 2. & (3) \end{aligned}$$

Some text books describe a variant of this algorithm, in which $v_0, \ldots, v_{t-1}$ are not computed and v is derived from the other polynomials as $v = v_t = \frac{r_t - r_0 u_t}{r_1}$. However, this variant is less efficient.

1.3 Adding divisors — Reduction step

To complete the addition described in the previous section we must find a reduced divisor equivalent to the divisor $\operatorname{div}(a, b)$ obtained. Notice that in the worst case a may have degree $2g$, namely for $\deg a_1 = \deg a_2 = g$ and $\deg d = 0$; in fact, this is also the typical case (see Section 3.2). We describe three different algorithms for the reduction step.

Gauß reduction

Let $a_0 = a$ and $b_0 = b$. Repeat the following computations for $k = 1, \ldots, t$, where $t \geq 0$ is the smallest index such that $\deg a_t \leq g$:

$$\begin{aligned} a_k &= \frac{f - b_{k-1}h - b_{k-1}^2}{a_{k-1}} \\ b_k &= (-h - b_{k-1}) \bmod a_k. \end{aligned}$$

Then $\operatorname{div}(a_t, b_t)$ is the reduced representation of $\operatorname{div}(a, b)$.

This algorithm was again described by Gauß for binary quadratic forms ([Gau01], Article 171). The case of hyperelliptic curves over fields of odd characteristic was treated by Artin ([Art24], §10); for a proof in the general case, see [MWZ98].

Cantor reduction

Gauß reduction amounts to replacing the divisor $\operatorname{div}(a, b)$ by the equivalent divisor $\overline{\operatorname{div}(b - Y) - \operatorname{div}(a, b)}$, which usually has a degree smaller by 2 (see Section 3.2). Cantor observed in [Can87] that it is possible to use an appropriate divisor $\operatorname{div}(c - dY)$ instead of $\operatorname{div}(b - Y)$. We describe his concept generalised to arbitrary characteristic.

The idea is to compute polynomials c and d such that $c = \lambda a + db$, $\deg c \leq \frac{\deg a + g}{2} \leq \frac{3g}{2}$ and $\deg d \leq \frac{\deg a - g - 1}{2} \leq \frac{g-1}{2}$. Set $s = \gcd(c, d) = \gcd(a, d)$, $a' = \frac{a}{s}$, $c' = \frac{c}{s}$ and $d' = \frac{d}{s}$. Let

$$\overline{a} = \frac{d'^2 f - c'd'h - c'^2}{a'},$$

$\overline{d}$ such that $d'\overline{d} \equiv 1 \bmod \overline{a}$ and

$$\overline{b} = (-h - \overline{d}c') \bmod \overline{a}.$$

Then $\operatorname{div}(a, b)$ is equivalent to

$$\operatorname{div}(\overline{a}, \overline{b}) + \operatorname{div}(s, b \bmod s),$$

and the latter sum can be computed using the composition algorithm of Section 1.2.

In practice, a different arrangement allows to compute s and $\overline{d}$ simultaneously by one application of the extended Euclidian algorithm. Let

$$\tilde{a} = \frac{d^2 f - cdh - c^2}{a}.$$

Determine $s = \gcd(\tilde{a}, d) = u\tilde{a} + \tilde{d}d$. If $\deg s \geq 1$, let $\overline{a} = \frac{\tilde{a}}{s}$ and $\overline{d} = \tilde{d}$, otherwise let $\overline{a} = \tilde{a}$ and $\overline{d} = s^{-1}\tilde{d}$. Then continue as before.

If c and d with the above properties exist, then we can closely follow Cantor's argumentation to show the validity of this approach. The equation $a = sa'$ implies $\operatorname{div}(a, b) = \operatorname{div}(s, b \bmod s) + \operatorname{div}(a', b)$, and

$$\begin{aligned}
\operatorname{div}(a', b) &= \gcd(\operatorname{div} a', \operatorname{div}(b - Y)) \\
&= \gcd(\operatorname{div} a', \operatorname{div}(d'b - d'Y)) \text{ because } a' \text{ and } d' \text{ are coprime} \\
&= \gcd(\operatorname{div} a', \operatorname{div}(c' - d'Y)) \text{ since } c' \equiv d'b \pmod{a'}.
\end{aligned}$$

The latter divisor is equivalent to $\overline{\operatorname{div}(c' - d'Y) - \gcd(\operatorname{div} a', \operatorname{div}(c' - d'Y))}$, and following the proof of Theorem 51 in [MWZ98] it is easy to see that this divisor is

$$\begin{aligned}
\gcd(\operatorname{div} \overline{a}, \operatorname{div}(c' + d'(Y + h))) &= \gcd(\operatorname{div} \overline{a}, \operatorname{div}(-\overline{d}c' - \overline{d}d'(Y + h))) \\
&\quad \text{because } \overline{a} \text{ and } \overline{d} \text{ are coprime} \\
&= \gcd(\operatorname{div} \overline{a}, \operatorname{div}(-\overline{d}c' - (Y + h))) \\
&\quad \text{because } \overline{d}d' \equiv 1 \pmod{\overline{a}} \\
&= \operatorname{div}(\overline{a}, -h - \overline{d}c').
\end{aligned}$$

However, c and d with the degree bounds imposed may not exist. Observe that c is a multiple of the greatest common divisor of a and b, and if $\deg(\gcd(a, b)) > \frac{\deg a + g}{2}$, then no c can keep the intended bound, and the best choice for c would seem to be $\gcd(a, b)$. But even with this choice the reduction process may fail; in fact, it may even increase the degree of the divisor, as illustrated by the following, fairly general example.

Suppose that $f = \prod_{i=1}^{g-1}(X - x_i) f_1$ with distinct $x_i \in K$ for $i = 1, \ldots, g-1$, so that $P_i = (x_i, 0) \in H$ for $i = 1, \ldots g - 1$. Let $a_1 = \prod_{i=1}^{g-1}(X - x_i)$ and $Q = (x_Q, y_Q)$ and $R = (x_R, y_R)$ two further points on H with coordinates in K, $x_1, \ldots, x_{g-1}$, x_Q and x_R all distinct from each other, $y_Q \neq 0$, $y_R \neq 0$ and $\frac{y_Q}{a_1(x_Q)} \neq \frac{y_R}{a_1(x_R)}$. Let $a = a_1(X - x_Q)(X - x_R)$ and $b = a_1 l$, where l is the unique linear polynomial such that $l(x_Q) = \frac{y_Q}{a_1(x_Q)}$ and $l(x_R) = \frac{y_R}{a_1(x_R)}$. Then

$$\sum_{i=1}^{g-1} P_i + Q + R = \operatorname{div}(a, b)$$

is semireduced of degree $g + 1$. The choice for c with lowest possible degree is

$$c = \gcd(a, b) = a_1 = \lambda a + db$$

with uniquely determined $\lambda \in K^\times$ and $d \in K[X]$ linear. If $s = \gcd(a_1, d) = 1$, then

$$\overline{a} = \frac{d^2 f_1 - dh - a_1}{(X - x_Q)(X - x_R)},$$

where $\deg(d^2 f_1) = g + 4$, $\deg(dh + a_1) \le g + 1$ and hence $\deg(\mathrm{div}(\overline{a}, \overline{b})) = \deg \overline{a} = g + 2 > \deg a$. Otherwise, the degree of $\mathrm{div}(\overline{a}, \overline{b})$ is smaller by $\deg s = 1$, but this is compensated by the addition of $\mathrm{div}(s, b \bmod s)$.

In case of failure, wich is extremely unlikely as shown by the analysis of Section 3.2, one must choose another reduction algorithm. To check whether polynomials c and d with the desired degrees do exist and to compute them in this case, one can use the extended Euclidian algorithm applied to a and b. With the notation of Section 1.2, if there is an index $t' \le t$ such that $\deg r_{t'} \le \frac{\deg a + g}{2}$, then $c = r_{t'}$ and $d = v_{t'}$ are suitable choices. Otherwise non-existence is shown.

Lagrange reduction

The most costly steps in the Gauß reduction procedure are the computations of the a_k, each involving one multiplication and one division of rather high degree polynomials. In the original reduction algorithm each step is independent of the previous one. However, as soon as one reduction step has been carried out, the formula for a_k can be rewritten using information from the previous step. The following algorithm was given for hyperelliptic curves over a field of odd characteristic by Paulus and Stein ([PS98]), who attribute the explicit formulae in the setting of a quadratic number field to Tenner. The main idea can be traced back to Lagrange ([Gra73], Théoreme II and Corollaire 1). Again we provide a generalised version for arbitrary characteristic.

Let $a_0 = a$ and $b_0 = b$, and compute a sequence (a_k, b_k) for $k = 1, \ldots, t$ by the following formulae, where $t \ge 0$ is the smallest index such that $\deg a_t \le g$:

$$\begin{aligned} a_1 &= \frac{f - b_0 h - b_0^2}{a_0} \\ -b_0 - h &= q_1 a_1 + b_1 \quad \text{with } \deg b_1 < \deg a_1 \end{aligned}$$

For $k \ge 2$:

$$\begin{aligned} a_k &= a_{k-2} + q_{k-1}(b_{k-1} - b_{k-2}) \\ -b_{k-1} - h &= q_k a_k + b_k \quad \text{with } \deg b_k < \deg a_k. \end{aligned}$$

Proof of correctness: We show that the sequence of divisors generated by the algorithm is exactly the same as in the Gauß reduction algorithm. To this purpose it is sufficient to show that

$$a_k = \frac{f - b_{k-1} h - b_{k-1}^2}{a_{k-1}},$$

which implies that also b_k is the same polynomial as in the usual Gauß reduction. We proceed by induction on k, the assertion being obviously true for $k = 1$. Hence let $k \ge 1$.

$$\begin{aligned} a_{k+1} a_k &= a_{k-1} a_k + q_k a_k (b_k - b_{k-1}) \\ &= f - b_{k-1} h - b_{k-1}^2 - (b_{k-1} + h + b_k)(b_k - b_{k-1}) \\ &\ \text{by the induction hypothesis and the construction of } q_k \\ &= f - b_k h - b_k^2, \end{aligned}$$

which proves the assertion. □

Notice that except for the first step the computation of a_k needs only one polynomial multiplication and no polynomial division, and that q_k is a biproduct of the computation of b_k.

1.4 Cryptographic setting

Our aim is to compare hyperelliptic cryptosystems of varying genus which offer the same level of security, and we are interested in the average time needed for adding two elements of the Jacobians.

As is the case for elliptic curves, no general subexponential algorithm is known for computing discrete logarithms in hyperelliptic Jacobians. Hence the only parameter determining the security of a hyperelliptic cryptosystem is the cardinality of the Jacobian group, which should be large enough to avoid square root attacks like Pollard's ρ-method or Shanks's baby-step giant-step algorithm (and also have a large prime factor to circumvent the Pohlig–Hellman attack, which we will not take into account here as there is no simple relation between the prime factors of the group order and the defining polynomials h and f).

The following theorem relates the genus g, the cardinality q of the base field and the cardinality of the Jacobian:

Theorem 1 (Weil) *Let $J(H)$ be the Jacobian of a hyperelliptic curve H of genus g defined over $\mathbb{F}_q$. Then*

$$(\sqrt{q}-1)^{2g} \leq |J(H)| \leq (\sqrt{q}+1)^{2g}.$$

Hence we fix a group size C according to the desired level of security ($C \approx 10^{50}$ is generally considered adequate), and compare hyperelliptic Jacobians which satisfy $q^g \approx C$, or equivalently $g \log q \approx c := \log C$ (in the above example, $c \approx 115$). So if g increases, then q decreases, the polynomials involved have higher degrees and smaller coefficients, and it is a priori unclear what would be the optimal value for g to obtain the fastest arithmetic.

However, if the genus becomes too large compared to the field size, then subexponential algorithms for taking discrete logarithms have been devised. In the odd characteristic case, an algorithm with conjectured subexponential running time was first provided by Adleman, DeMarrais and Huang in [ADH94], an algorithm with rigorously proven running time was presented by Müller, Stein and Thiel in [MST99], with details filled in by the author and Stein ([ES00]). The author generalised this algorithm to even characteristic and gave a more detailed analysis in [Eng99a]. The running time of the algorithm grows with decreasing ratio $g/\log q$, whence we assume that g is "not too big" and q "not too small". Thus, we may also assume conventional arithmetic for polynomial computation.

2 The Extended Euclidian Algorithm on Polynomials

The composition step of Section 1.2 and the Cantor reduction of Section 1.3 rely on the extended Euclidian algorithm, or to be more precise on its half-extended version, since one of the multipliers is not needed. Consequently we have to analyse the average complexity of the Euclidian algorithm. We hereby concentrate on the case of two polynomials, since Lemma 6 shows that these are usually coprime and hence no further computations involving a third polynomial are needed.

While there is an abundant literature on the Euclidian algorithm on integers (see [Knu81], Section 4.5.3 and the references therein), I am aware of only a few articles dealing with its average complexity when applied to polynomials: Arnold and John Knopfmacher analyse the average number of polynomial divisions ([KK88]), while Ma and von zur Gathen present the average number of field operations for different variants of the algorithm over $\mathbb{F}_2[X]$ ([MG90]). In [Ma87], Chapter 3, Ma gives a more comprehensive account of the Euclidian algorithm over any finite field, including the average complexity. However, neither of them treats the computation of the multipliers. Ma uses a rather intricate counting argument, which seems difficult to generalise to the extended algorithm. Instead we prefer to explicitly compute the probability distributions of the random variables involved, hereby presenting alternate proofs for some of Ma's results.

2.1 Average complexity

Throughout this section we assume that we are given two polynomials r_0 and r_1 over a finite field $\mathbb{F}_q$ of fixed degrees $d_0 \geq d_1 \geq 1$, respectively, and that these input data are uniformly distributed over all polynomials of degrees d_0 and d_1. A run of the extended Euclidian algorithm as presented in Section 1.2 creates sequences $(r_0, \dots, r_t)$, $(u_0, \dots, u_t)$ and $(v_0, \dots, v_t)$ of elements of $\mathbb{F}_q[X]$, where t and the r_k, u_k and v_k are random variables. Denote the random variable $\deg r_k$ by d_k with the convention $d_k = -\infty$ for $k > t$. By a slight abuse of notation, we let d_t denote the degree of the last non-zero remainder and d_{t-1} the degree of the next to last one.

We are interested in the average number of arithmetic operations in $\mathbb{F}_q$ needed for carrying out the algorithm; since additions and subtractions are essentially for free, we restrict our attention to the number of multiplications and inversions in $\mathbb{F}_q$. The following theorem states the main result of this section:

Theorem 2 *Let r_0 and r_1 be uniformly distributed over the polynomials over $\mathbb{F}_q$ of degrees $d_0 \geq d_1 \geq 1$, respectively. Denote by $E_m(d_0, d_1)$ and $E_i(d_0, d_1)$ the expected number of multiplications and inversions in $\mathbb{F}_q$ performed by the non-extended Euclidian algorithm, and by $E^u(d_0, d_1)$ and $E^v(d_0, d_1)$ the expected number of additional multiplications needed to compute the multipliers u and v, respectively. Then all these quantities are polynomials in $1/q$; more precisely,*

$$
\begin{aligned}
&E_m(d_0,d_1)(q-1)^2 q^{d_1}\\
&\quad= \; (d_0d_1 + d_0 + d_1 - 1)\, q^{d_1+2} - \left(2d_0d_1 + \tfrac{1}{2}d_1^2 + 2d_0 + \tfrac{7}{2}d_1 - 3\right) q^{d_1+1}\\
&\qquad + \left(d_0d_1 + d_1^2 + d_0 + 3d_1 - 2\right) q^{d_1} - \left(\tfrac{1}{2}d_1^2 + \tfrac{1}{2}d_1 - 1\right) q^{d_1-1}\\
&\qquad + q^2 - 2q\\
&E_i(d_0,d_1) = d_1 - \frac{d_1 - 1}{q}\\
&E^u(d_0,d_1)(q-1)^2 q^{d_1}\\
&\quad= \; \left(d_1^2 - d_1 - 2\right) q^{d_1+2} - \left(\tfrac{5}{2}d_1^2 + \tfrac{5}{2}d_1 - 10\right) q^{d_1+1} + \left(2d_1^2 + 5d_1 - 2\right) q^{d_1}\\
&\qquad - \left(\tfrac{1}{2}d_1^2 + \tfrac{3}{2}d_1\right) q^{d_1-1} + \left(\tfrac{1}{2}d_1^2 + \tfrac{3}{2}d_1\right) q^3 - \left(2d_1^2 + 5d_1 - 2\right) q^2\\
&\qquad + \left(\tfrac{5}{2}d_1^2 + \tfrac{5}{2}d_1 - 10\right) q - (d_1^2 - d_1 - 2)\\
&E^v(d_0,d_1)(q-1)^2 q^{d_1}\\
&\quad= \; (2d_0 - d_1)(d_1 - 1) q^{d_1+2} - \left(5d_0d_1 - \tfrac{5}{2}d_1^2 - 2d_0 + \tfrac{9}{2}d_1 - 7\right) q^{d_1+1}\\
&\qquad + \left(4d_0d_1 - 2d_1^2 + 5d_1 - 1\right) q^{d_1} - \left(d_0d_1 - \tfrac{1}{2}d_1^2 + \tfrac{3}{2}d_1\right) q^{d_1-1}\\
&\qquad + \left(d_0d_1 - \tfrac{3}{2}d_1^2 + 2d_0 - \tfrac{7}{2}d_1\right) q^2 - (2d_0d_1 - 3d_1^2 + 2d_0 - 2d_1 + 7) q\\
&\qquad + \left(d_0d_1 - \tfrac{3}{2}d_1^2 + \tfrac{3}{2}d_1 + 1\right)
\end{aligned}
$$

We certainly need to know the arithmetic complexity of the basic polynomial operations. Let f_1 and f_2 be polynomials of degree $n_1 \geq n_2$, respectively. The number of field multiplications needed to compute $f_1 f_2$ depends on the number of non-zero coefficients of f_1 and f_2. However, observing that the number of non-zero non-leading coefficients of a random polynomial of degree n over $\mathbb{F}_q$ is Binomial$(n, 1 - 1/q)$–distributed, we assume that all polynomials involved have no zero coefficients, which biases the results only negligibly since q is big. Now multiplying f_1 and f_2 takes

$(n_1+1)(n_2+1)$ field multiplications, and a polynomial division with remainder of f_1 by f_2 takes one inversion for the leading coefficient of f_2 and $(n_2+1)(n_1-n_2+1)$ multiplications.[1]

Noticing that the final division by r_t can be omitted if $d_t = 0$, we see that the non-extended Euclidian algorithm requires

$$\sum_{k=1,\ldots,t\,:\,d_k\neq 0} (d_k+1)(d_{k-1}-d_k+1)$$

multiplications and $|\{k=1,\ldots,t \,:\, d_k \neq 0\}|$ inversions. The computation of the u_k needs additional

$$\sum_{k=4}^{t}(d_{k-2}-d_{k-1}+1)(d_1-d_{k-2}+1)$$

and the computation of the v_k

$$\sum_{k=3}^{t}(d_{k-2}-d_{k-1}+1)(d_0-d_{k-2}+1)$$

multiplications. Notice that $u_2 = 1$, $u_3 = -q_2$ and $v_2 = -q_1$ are given for free, and that the degrees of the q_k, u_k and v_k can be expressed in terms of the d_k by Formulae (1) to (3).

Then the expected numbers of field operations are given by

$$\begin{aligned}
E_m(d_0,d_1) &= (d_1+1)(d_0-d_1+1) + \sum_{\mu=1}^{\infty} P(d_2=\mu)(\mu+1)(d_1-\mu+1) \\
&\quad + \sum_{k=3}^{\infty}\sum_{\mu=0}^{\infty}\sum_{\nu=1}^{\infty} P(d_{k-1}=\mu d_k=\nu)(\nu+1)(\mu-\nu+1) && (4)\\
E_i(d_0,d_1) &= \sum_{\tau=1}^{\infty} \tau P(t=\tau) - P(\deg\gcd(d_0,d_1))=0) && (5)\\
E^u(d_0,d_1) &= \sum_{k=4}^{\infty}\sum_{\mu=0}^{\infty}\sum_{\nu=0}^{\infty} P(d_{k-2}=\mu d_{k-1}=\nu d_k\neq-\infty) && (6)\\
&\quad (\mu-\nu+1)(d_1-\mu+1) \\
E^v(d_0,d_1) &= \sum_{k=3}^{\infty}\sum_{\mu=0}^{\infty}\sum_{\nu=0}^{\infty} P(d_{k-2}=\mu d_{k-1}=\nu d_k\neq-\infty) && (7)\\
&\quad (\mu-\nu+1)(d_0-\mu+1)
\end{aligned}$$

The condition "$d_k \neq \infty$" in the expressions for E^u and E^v assures that the summation stops with $k = t$ instead of $k = t + 1$.

Now, to prove Theorem 2, it is sufficient to determine the various probabilities occuring in the above formulae. The key observation is the following lemma:

[1] This differs from Ma's assumption of needing $n_2(n_1-n_2+1)$ multiplications and n_1-n_2+1 divisions ([Ma87], Fact 3.3). Taking into acount that divisions are more costly in a finite field than multiplications and observing that all divisions are actually by the leading coefficient of f_2, it is more efficient to invert this coefficient once and for all and to replace the divisions by multiplications.

Lemma 3 *Let $f_1 \in \mathbb{F}_q[X]$ be uniformly distributed over the polynomials of degree α_1, $f_2 \in \mathbb{F}_q[X]$ be fixed of degree $\alpha_2 \leq \alpha_1$ and f_3 the remainder of f_1 divided by f_2. Then f_3 is uniformly distributed over the polynomials of degree less than α_2. It follows for $\mu > \nu \geq 0$ that*

$$
\begin{aligned}
P(d_k = \nu | d_{k-1} = \mu) &= \frac{\#\ \textit{polynomials of degree}\ \nu}{\#\ \textit{polynomials of degree less than}\ \mu} \\
&= (q-1)q^{\nu-\mu}, \\
P(d_k = -\infty | d_{k-1} = \mu) &= q^{-\mu}.
\end{aligned}
$$

Proof: Note that for fixed f_2 the map $f_1 \mapsto f_3$ is an epimorphism of the additive group of polynomials of degree α_1 onto the additive group of polynomials of degree less than α_2. Since any preimage under a group epimorphism has the same cardinality, this proves the assertion. □

These conditional probabilities can be used to determine the total probability of d_k admitting specific values:

Lemma 4

$$
\begin{aligned}
P(d_k = -\infty) &= \begin{cases} \frac{1}{q^{d_1}} \sum\limits_{j=0}^{k-2} \binom{d_1}{j} (q-1)^j & \textit{for } 2 \leq k \leq d_1 + 1 \\ 1 & \textit{for } k \geq d_1 + 2 \end{cases} \\
P(d_k = \nu) &= \begin{cases} \frac{(q-1)^{k-1}}{q^{d_1}} q^{\nu} \binom{d_1 - \nu - 1}{k-2} & \textit{for } \begin{array}{l} 2 \leq k \leq d_1 + 1, \\ 0 \leq \nu \leq d_1 - k + 1 \end{array} \\ 0 & \textit{otherwise} \end{cases}
\end{aligned}
$$

Proof: Observe first that $d_k \leq d_{k-1} - 1$ implies $d_k \leq d_1 - k + 1$ for $k \geq 2$; this proves the trivial parts of the assertion. We now restrict ourselves to the interesting cases and proceed by induction for $2 \leq k \leq d_1$, noting that the assertions for $k = 2$ follow directly from Lemma 3.

$$
\begin{aligned}
&P(d_{k+1} = -\infty) \\
&= \sum_{\mu=0}^{d_1-k+1} P(d_k = \mu) P(d_{k+1} = -\infty | d_k = \mu) + P(d_k = -\infty) \\
&= \sum_{\mu=0}^{d_1-k+1} \frac{(q-1)^{k-1}}{q^{d_1}} q^{\mu} \binom{d_1 - \mu - 1}{k-2} \frac{1}{q^{\mu}} + \frac{1}{q^{d_1}} \sum_{j=0}^{k-2} \binom{d_1}{j} (q-1)^j \\
&\qquad \text{by the induction hypothesis and Lemma 3} \\
&= \frac{(q-1)^{k-1}}{q^{d_1}} \underbrace{\sum_{\mu=k-2}^{d_1-1} \binom{\mu}{k-2}}_{=\binom{d_1}{k-1}} + \frac{1}{q_{d_1}} \sum_{j=0}^{k-2} \binom{d_1}{j} (q-1)^j
\end{aligned}
$$

Let now be $0 \leq \nu \leq d_1 - k$.

$$\begin{aligned}
P(d_{k+1} = \nu) &= \sum_{\mu=\nu+1}^{d1-k+1} P(d_k = \mu) P(d_{k+1} = \nu | d_k = \mu) \\
&= \sum_{\mu=\nu+1}^{d1-k+1} \frac{(q-1)^{k-1}}{q^{d_1}} q^{\mu} \binom{d_1 - \mu - 1}{k-2} \frac{(q-1)q^{\nu}}{q^{\mu}} \\
&= \frac{(q-1)^k}{q^{d_1}} q^{\nu} \sum_{\mu=k-2}^{d_1-\nu-2} \binom{\mu}{k-2} \\
&= \frac{(q-1)^k}{q^{d_1}} q^{\nu} \binom{d_1 - \nu - 1}{k-1}
\end{aligned}$$

□

The probability distribution of t has already been determined by Ma ([Ma87], Lemma 3.11); we give a simple proof in our setting:

Lemma 5

$$P(t = k) = \begin{cases} \binom{d_1}{k-1} \frac{(q-1)^{k-1}}{q^{d_1}} & \textit{for } 1 \leq k \leq d_1 + 1 \\ 0 & \textit{otherwise} \end{cases}$$

Proof: From the remark at the beginning of the proof of Lemma 4 it is clear that $t \leq d_1 + 1$. Hence, let $1 \leq k \leq d_1 + 1$. Then

$$\begin{aligned}
P(t = k) &= P(d_{k-1} \neq -\infty) P(d_k = -\infty | d_{k-1} \neq -\infty) \\
&= \sum_{\mu=0}^{\infty} P(d_{k-1} = \mu) P(d_k = -\infty | d_{k-1} = \mu),
\end{aligned}$$

and substituting the results of Lemmata 3 and 4 into this formula yields the desired expression. □

Proof of Theorem 2: Lemmata 3 and 4 allow us to compute the joint probability distribution of sequences of remainder degrees; for instance,

$$\begin{aligned}
&P(d_{k-2} = \mu d_{k-1} = \nu d_k \neq \infty) \\
&= P(d_{k-2} = \mu) P(d_{k-1} = \nu | d_{k-2} = \mu)(1 - P(d_k = -\infty | d_{k-1} = \nu)).
\end{aligned}$$

Substituting the results of the lemmata into Equations (4) to (7) yields the desired formulae after some tedious calculations. Note then that the numerator of E_m has a double zero in 1, so it is divisible by $(q-1)^2$. Comparing degrees of the numerator and denominator shows that E_m is indeed a polynomial in $1/q$. Similar reasonings apply to E^u and E^v, and E_i is trivially a polynomial in $1/q$. □

Lemma 3 shows that with probability $1 - \frac{1}{q}$, we have $d_k = d_{k-1} - 1$ for $1 \leq d_{k-1} \leq d_1$. Hence the leading coefficients of E_m, E_i, E^u and E^v could alternatively be determined by examining this typical case.

While it is well known that the asymptotic worst-case complexity of the (extended) Euclidian algorithm is quadratic for the number of multiplications and linear for the number of inversions, Theorem 2 shows that this still holds in the average case and for small degree polynomials.

2.2 Some more probabilities

For the composition step in Section 1.2 we have to carry out further computations with the greatest common divisor and the multipliers, so that the probability distributions of their respective degrees are of interest. Recall from Section 2.1 that d_t denotes the degree of the greatest common divisor and that the multiplier degrees are related to d_{t-1} by (2) and (3).

Lemma 6

$$
P(d_{t-1} = \mu d_t = \nu) = \begin{cases} \dfrac{1}{q^{d_1}} & \textit{for } \mu = d_0 \textit{ and } \nu = d_1 \\ \dfrac{q-1}{q^{d_1}} & \textit{for } \mu = d_1 \textit{ and } 0 \leq \nu < d_1 \\ \dfrac{(q-1)^2}{q^{\mu+1}} & \textit{for } 1 \leq \mu < d_1 \textit{ and } 0 \leq \nu < \mu \\ 0 & \textit{otherwise} \end{cases}
$$

$$
P(d_{t-1} = \mu) = \begin{cases} \dfrac{1}{q^{d_1}} & \textit{for } \mu = d_0 \\ d_1 \dfrac{q-1}{q^{d_1}} & \textit{for } \mu = d_1 \\ \mu \dfrac{(q-1)^2}{q^{\mu+1}} & \textit{for } 1 \leq \mu < d_1 \\ 0 & \textit{otherwise} \end{cases}
$$

$$
P(d_t = \nu) = \begin{cases} \dfrac{1}{q^{d_1}} & \textit{for } \nu = d_1 \\ \dfrac{q-1}{q^{\nu+1}} & \textit{for } 0 \leq \nu < d_1 \\ 0 & \textit{otherwise} \end{cases}
$$

Proof: Observing that

$$
\begin{aligned}
& P(d_{t-1} = \mu d_t = \nu) \\
& = \sum_{k=1}^{\infty} P(d_{k-1} = \mu) P(d_k = \nu | d_{k-1} = \mu) P(d_{k+1} = -\infty | d_k = \nu), \\
& P(d_{t-1} = \mu) = \sum_{\nu=0}^{\infty} P(d_{t-1} = \mu d_t = \nu) \quad \text{and} \\
& P(d_t = \nu) = \sum_{\mu=0}^{\infty} P(d_{t-1} = \mu d_t = \nu),
\end{aligned}
$$

this is an easy consequence of Lemmata 3 and 4. □

The result for d_t has already been proved by Ma and von zur Gathen ([MG90], Proposition 2.5). Letting $\nu = 0$, the lemma shows that two random polynomials are coprime with probability $1 - 1/q$.

A further result is needed to analyse Cantor's reduction algorithm of Section 1.3; the reduction involves the computation of $\gcd(r_k, v_k)$, and these two polynomials are not independently distributed, so that the previous lemma cannot be applied. However, it is possible to give an estimate of the probability that r_k and v_k are coprime.

Lemma 7 *Let $k \geq 2$ and $d_{k-1} > 0$. Then*

$$P(\deg\gcd(r_k, v_k)) = 0) \geq 1 - \frac{d_0 - 1}{q}.$$

Proof: Lemma 3 shows that r_k is uniformly distributed over the polynomials of degree at most $d_{k-1} - 1 \geq 0$. Hence,

$$\begin{aligned} P(\deg\gcd(r_k, v_k)) = 0) &\geq 1 - \sum_{\pi | v_k,\ \pi \text{ irreducible}} P(\pi | r_k) \\ &= 1 - \sum_{\pi | v_k,\ \pi \text{ irreducible, } \deg \pi \leq d_{k-1} - 1} \frac{1}{q^{\deg \pi}} \\ &\quad \text{because of the uniform distribution of } r_k \\ &\geq 1 - \frac{\deg v_k}{q} \geq 1 - \frac{d_0 - 1}{q} \end{aligned}$$

□

3 The Efficiency of Hyperelliptic Arithmetic

In this section we analyse the average complexity of adding two randomly chosen elements of the Jacobian of a hyperelliptic curve. To make use of the results of the previous section, we must check whether our assumptions on the probability distributions hold, which we do in Section 3.1. Then in Section 3.2 we provide the average number of field operations for various settings, like adding two points or doubling one point in odd or even characteristic. Finally in Section 3.3 we take into account that operations in different fields take a different amount of time.

3.1 Heuristic assumptions on the probabilities

To apply the results of the previous section we have to argue that in a random divisor $\operatorname{div}(a, b)$ the polynomials a and b are uniformly distributed over all polynomials of fixed degrees. Clearly this is not true if we fix a curve, since then most of the pairs (a, b) do not define a divisor. Hence, we must also let the curve vary. Even then a and b are not exactly uniformly distributed, but we give a heuristic argument that the deviation is sufficiently small.

Heuristic 8 *Let the hyperelliptic curve H of fixed genus g be chosen randomly according to a uniform distribution on the defining pairs of suitable polynomials (h, f). If $\operatorname{div}(a, b)$ is a uniformly selected element of $J(H)$, then we can assume that a varies uniformly over all polynomials of degree g and b over all polynomials of degree $g-1$. Likewise, if $\operatorname{div}(a_1, b_1)$ and $\operatorname{div}(a_2, b_2)$ are two uniformly selected elements of $J(H)$, then we can assume that a_1 and a_2 vary uniformly and independently over all polynomials of degree g and b_1 and b_2 over all polynomials of degree $g - 1$.*

Justification: We examine only the case of two divisors because all arguments hold and slightly simplify if only one divisor is considered. We make two simplifying assumptions, which account for the heuristic not being a theorem: First, we assume that any two polynomials h of degree at most g and f monic of degree $2g + 1$ define a hyperelliptic curve, not taking into account that a negligible proportion of these curves is singular. Hence there are q^{3g+2} curves altogether. Second, we assume that all Jacobians have cardinality q^g (cf. Theorem 1), so that $(q-1)q^g$ pairs of polynomials (a,b) define a Jacobian element because we do not require a to be monic.

Let (a_1,b_1,a_2,b_2) be a quadruple of polynomials such that $\deg b_1 < \deg a_1 \le g$ and $\deg b_2 < \deg a_2 \le g$. Assume for the moment that h is fixed; then there are $q^{2g+1-\deg \operatorname{lcm}(a_1,a_2)}$ curves admitting $\operatorname{div}(a_1,b_1)$ and $\operatorname{div}(a_2,b_2)$ simultaneously as elements of their Jacobians. Namely these curves correspond to the monic solutions of degree $2g+1$ of the equations $f \equiv b_1^2 + hb_1 \pmod{a_1}$ and $f \equiv b_2^2 + hb_2 \pmod{a_2}$, which by the Chinese Remainder Theorem translate into a single equation modulo $\operatorname{lcm}(a_1,a_2)$. Hence if we also let h vary there are $q^{3g+2-\deg \operatorname{lcm}(a_1,a_2)}$ curves containing both of the divisors in their Jacobians, and the probability of selecting one of them is $q^{-\deg \operatorname{lcm}(a_1,a_2)} = q^{-\deg a_1 - \deg a_2 + \deg \gcd(a_1,a_2)}$. Once such a curve is fixed, the conditional probability of choosing (a_1,b_1,a_2,b_2) is given by $\left(\frac{1}{(q-1)q^g}\right)^2$. Thus, the quadruple occurs with a total probability of $q^{-2g-\deg a_1 - \deg a_2 + \deg \gcd(a_1,a_2)}(q-1)^{-2}$.

Since we have seen in Lemma 6 that two polynomials are coprime with probability $1-1/q$ and we assumed q to be large, we can drop the term "$\deg \gcd(a_1,a_2)$" in the exponent. Then all the quadruples of polynomials with $\deg a_1 = \deg a_2 = g$ have the same probability of $q^{-4g}(q-1)^{-2}$. Hence, the event $\deg a_1 = \deg a_2 = g$ and $\deg b_1 = \deg b_2 = g-1$ occurs with probability

$$\frac{(q-1)^4 q^{4g-2}}{(q-1)^2 q^{4g}} = \left(1 - \frac{1}{q}\right)^2,$$

which is very close to 1. □

3.2 The average number of field operations

In this section we determine the average number of field multiplications and inversions performed during the different algorithms of Sections 1.2 and 1.3. We give correct asymptotic formulae for $q \to \infty$, i.e., we do not take into account events that have probability in $\frac{1}{q}O(1)$.

We have to distinguish several cases. In odd characteristic we can assume $h = 0$, while in even characteristic we have $h \ne 0$ and the exact complexities depend on $\deg h$; a complete description would comprise the cases $h = 1$, $h \in \mathbb{F}_q^\times \backslash \{1\}$, $\deg h = g$, $\deg h = g-1$ and $1 \le \deg h \le g-2$. In order to keep the presentation concise we concentrate on two cases especially attractive for implementation, namely the curves with $h = 1$ and the curves with $\deg h = 1$, in which case an affine change of variables results in $h = X$.

Concerning the polynomial arithmetic, it should be noted that squaring a polynomial is more efficient than multiplying two distinct polynomials. In characteristic 2, squaring a polynomial of degree n takes $n+1$ squarings in the underlying field. Assuming that the field arithmetic is implemented using normal bases, these squarings are essentially for free, hence we do not count them. In odd characteristic, we can write

$$\left(\sum_{i=0}^{n} a_i X^i\right)^2 = \sum_{i=0}^{n} a_i^2 X^{2i} + \sum_{i=0}^{n}\sum_{j=0}^{i-1} (a_i a_j + a_i a_j) X^{i+j}$$

and need $(n+1) + \frac{1}{2}n(n+1) = \frac{1}{2}(n+1)(n+2)$ field multiplications.

All complexities given are quadratic or cubic polynomials in g with fixed leading coefficient, but the other coefficients may vary according to the arrangement of the computations. Exchanging two multiplications, for instance, may already have an impact. Since all proofs are very alike, we present only one proof in detail and proceed to simply outline the order of the computations in the subsequent theorems, leaving the details to the reader. While much care has been taken to find the optimal arrangements of computations, it is possible that sometimes better ones exist.

Composition

According to Section 3.1, we assume that the composition algorithm is applied to uniformly distributed polynomials a_1, a_2, b_1 and b_2 of degrees g, g, $g-1$ and $g-1$, respectively.

Theorem 9 *On average, two distinct reduced divisors can be composed with*

- $8g^2 + 5g - 2 + \frac{1}{q}O(g^2)$ *multiplications and* $g + 2 + \frac{1}{q}O(g)$ *inversions for* p *odd;*
- $7g^2 + 7g - 1 + \frac{1}{q}O(g^2)$ *multiplications and* $g + 1 + \frac{1}{q}O(g)$ *inversions for* $p = 2$, $h \in \{1, X\}$.

On average, a reduced divisor can be composed with itself with

- $7g^2 + 7g - 1 + \frac{1}{q}O(g^2)$ *multiplications and* $g + 1 + \frac{1}{q}O(g)$ *inversions for* p *odd;*
- $4g + 2$ *multiplications and one inversion for* $p = 2$, $h = 1$;
- $4g^2 + 5g + 2$ *multiplications and two inversions for* $p = 2$, $h = X$.

Proof: We first analyse the composition step for distinct divisors and assume that $p \neq 2$. The average number of multiplications during the computation of $\gcd(a_1, a_2)$ and u_1 is in $2g^2 + g - 3 + \frac{1}{q}O(g^2)$, the average number of inversions is in $g + \frac{1}{q}O(g)$ by Theorem 2. The greatest common divisor has, by Lemma 6, degree zero with probability $1 - \frac{1}{q}$. We can thus assume this case, since otherwise we apply the Euclidian algorithm a second time and all further computations are covered by the "$\frac{1}{q}O(g^2)$" and "$\frac{1}{q}O(g)$" terms. So we suppose $u_3 = 0$. Moreover, Lemma 6 shows that $\deg u_1 = g - 1$ is most likely. Compute d^{-1} and then $a = ((d^{-1})^2 a_1) a_2$ with $1 + (g+1) + (g+1)^2$ multiplications and $b_1 + ((d^{-1}u_1)(b_2 - b_1))a_1$ with $g + g^2 + (2g-1)(g+1)$ multiplications. Since the result is most likely of degree $3g - 2$ and a of degree $2g$, the reduction modulo a needs additional $(2g+1)(g-1)$ multiplications and one inversion, which proves the desired result.

The same reasoning applies to the case $p = 2$, but a different approach is preferable. Compute $d = \gcd(a_2, b_1 + b_2 + h)$, d^{-1} and u_3 with expected $2g^2 - 4 + \frac{1}{q}O(g^2)$ multiplications and $g + \frac{1}{q}O(g)$ inversions, and $a = ((d^{-1})^2 a_1) a_2$ with $(g+1) + (g+1)^2$ multiplications. Reduce f modulo a to obtain $\tilde{f}$ by $2(2g+1)$ multiplications and one inversion. Compute $\tilde{b} = b_1^2 + b_1 h$ for free since squaring and multiplying by $h \in \{1, X\}$ are free, then $b_1 + (d^{-1}u_3)(\tilde{f} + \tilde{b})$ with $g + 2g^2$ multiplications, noticing that the degree of u_3 is most likely $g - 1$ by Lemma 6 and the degree of $\tilde{f} + \tilde{b}$ most likely $2g - 1$ by Lemma 3. The final reduction modulo a needs again $(2g+1)(g-1)$ multiplications, but this time no inversion since the inverted leading coefficient of a is still known from the reduction of f. This approach needs fewer multiplications as soon as $g \geq 3$. For $g = 2$, it saves one inversion at the expense of two multiplications, which is usually faster (see the references in Section 3.3).

The same approach of computing $\gcd(a_1, 2b_1 + h)$ must be taken for doubling a point, which results in the other complexities given. Notice that a should be computed as $(d^{-1}a_1)^2$. For $p = 2$ and $h = X$, the values of d and u_3 are trivial to determine; the given complexity is then valid for the most probable case $\deg d = 0$, the case $d = X$ needs even fewer operations. If $p = 2$ and $h = 1$, then we even have $d = u_3 = 1$, which implies a further simplification. □

Gauß reduction

In order to apply the results of Section 2 and in accordance with Section 3.1 we assume that the composition algorithm has yielded a random semireduced divisor $\operatorname{div}(a, b)$ of degree $2g$, so that a and b are approximately uniformly distributed over all polynomials of degree $2g$ and $2g - 1$, respectively.

Theorem 10 *On average, Gauß reduction can be executed with*

- $\frac{7}{4}g^3 + \frac{33}{8}g^2 + \frac{1}{4}g + \frac{1}{q}O(g^4)$ *multiplications and* $\frac{1}{2}g + 1 + \frac{1}{q}O(g^2)$ *inversions for* p *odd,* g *even;*
- $\frac{7}{4}g^3 + \frac{39}{8}g^2 + \frac{9}{4}g + \frac{17}{8} + \frac{1}{q}O(g^4)$ *multiplications and* $\frac{1}{2}g + \frac{3}{2} + \frac{1}{q}O(g^2)$ *inversions for* p *odd,* g *odd;*
- $\frac{7}{6}g^3 + 3g^2 - \frac{1}{6}g + \frac{1}{q}O(g^4)$ *multiplications and* $\frac{1}{2}g + 1 + \frac{1}{q}O(g^2)$ *inversions for* $p = 2$, $h \in \{1, X\}$, g *even;*
- $\frac{7}{6}g^3 + \frac{7}{2}g^2 + \frac{4}{3}g + 2 + \frac{1}{q}O(g^4)$ *multiplications and* $\frac{1}{2}g + \frac{3}{2} + \frac{1}{q}O(g^2)$ *inversions for* $p = 2$, $h \in \{1, X\}$, g *odd.*

Proof: Assuming the notation of Section 1.3, Lemma 3 shows that with probability in $1 - \frac{1}{q}O(g)$ we have $t = \lceil \frac{g}{2} \rceil$, $\deg a_k = 2(g-k)$ and $\deg b_k = 2(g-k) - 1$ for $k < t$ and $\deg a_t = g$. Now some simple computations reveal the results above. □

Cantor reduction

If $g = 2$, then in the Cantor reduction step we have $c = b$ and $d = 1$, so that it equals Gauß reduction. Hence we assume $g \geq 3$.

Theorem 11 *If* $g \geq 3$, *Cantor reduction is successful with probability at least* $1 - \frac{1}{q^{g+1}}$. *On average, it can then be executed with*

- $11g^2 + 3g - 8 + \frac{1}{q}O(g^3)$ *multiplications and* $g + 1 + \frac{1}{q}O(g^2)$ *inversions for* p *odd,* g *even;*
- $11g^2 + 7g - 4 + \frac{1}{q}O(g^3)$ *multiplications and* $g + 2 + \frac{1}{q}O(g^2)$ *inversions for* p *odd,* g *odd;*
- $\frac{21}{2}g^2 + g - 9 + \frac{1}{q}O(g^3)$ *multiplications and* $g + 1 + \frac{1}{q}O(g^2)$ *inversions for* $p = 2$, $h \in \{1, X\}$, g *even;*
- $\frac{21}{2}g^2 + 6g - \frac{9}{2} + \frac{1}{q}O(g^3)$ *multiplications and* $g + 2 + \frac{1}{q}O(g^2)$ *inversions for* $p = 2$, $h \in \{1, X\}$, g *odd.*

Proof: Cantor reduction is successful at least when $\deg(\gcd(a, b)) \leq g$, which by Lemma 6 happens with probability $1 - \frac{1}{q^{g+1}}$. In this case, applying the extended Euclidian algorithm to a and b without computing the u_k and stopping as soon as the desired degree is reached yields c and d and requires

$$\sum_{k=2}^{t'} (d_{k-1} + 1)(d_{k-2} - d_{k-1} + 1) + \sum_{k=3}^{t'} (d_{k-2} - d_{k-1} + 1)(d_0 - d_{k-2} + 1)$$

multiplications and $t' - 1$ inversions, where by Lemma 3 the probability that all $d_k = 2g - k$ and $t' = \lceil \frac{g}{2} \rceil$ is in $1 - \frac{1}{q}O(g)$. Compute $\tilde{a}$ and apply the half-extended Euclidian algorithm to $\tilde{a}$, which is usually of degree g, and d, which is usually of degree $\lceil \frac{g}{2} \rceil - 1$, to compute the greatest common divisor s and $\tilde{d}$. By Lemma 7 we most probably have $\deg s = 0$, so that $\overline{a} = \tilde{a}$ and it is sufficient to compute $\overline{b} = (-h - \overline{d}(c \bmod \overline{a})) \bmod \overline{a}$ to obtain the reduced divisor $\operatorname{div}(\overline{a}, \overline{b})$. □

Lagrange reduction

Again the reduction process differs from Gauß reduction only for $g \geq 3$.

Theorem 12 *If $g \geq 3$, then on average Lagrange reduction can be executed with*

- $9g^2 - 5 + \frac{1}{q}O(g^3)$ *multiplications and* $\frac{1}{2}g + 1 + \frac{1}{q}O(g^2)$ *inversions for p odd, g even;*
- $9g^2 + g - 2 + \frac{1}{q}O(g^3)$ *multiplications and* $\frac{1}{2}g + \frac{3}{2} + \frac{1}{q}O(g^2)$ *inversions for p odd, g odd;*
- $7g^2 - g - 5 + \frac{1}{q}O(g^3)$ *multiplications and* $\frac{1}{2}g + 1 + \frac{1}{q}O(g^2)$ *inversions for $p = 2$, $h \in \{1, X\}$, g even;*
- $7g^2 - 2 + \frac{1}{q}O(g^3)$ *multiplications and* $\frac{1}{2}g + \frac{3}{2} + \frac{1}{q}O(g^2)$ *inversions for $p = 2$, $h \in \{1, X\}$, g odd.*

Proof: The analysis is straightforward since by Lemma 3 we have $t = \lceil \frac{g}{2} \rceil$, $\deg a_k = 2(g-k)$, $\deg b_k = 2(g-k) - 1$ and $\deg q_k = 1$ for $k \leq \lfloor \frac{g}{2} \rfloor$ with total probability in $1 - \frac{1}{q}O(g)$. If g is odd, then $\deg a_t = g$ and $\deg q_t = 0$ with overwhelming probability. □

Complete addition step

To estimate the number of field operations required for a full addition step we have to choose between the three reduction procedures. Comparison of the numbers in Theorems 10 to 12 yields the following result.

Theorem 13 *For large q, Lagrange reduction is the fastest reduction procedure.*

We can now use the results of Theorems 9, 10 and 12 to determine the number of field operations needed in the different cases.

Theorem 14 *On average, two distinct divisors can be added with the following numbers of field operations:*

	multiplications	*inversions*
$p \neq 2$, g *even*	$17g^2 + 5g - 7 + \frac{1}{q}O(g^3)$	$\frac{3}{2}g + 3 + \frac{1}{q}O(g^2)$
$p \neq 2$, g *odd*	$17g^2 + 6g - 4 + \frac{1}{q}O(g^3)$	$\frac{3}{2}g + \frac{7}{2} + \frac{1}{q}O(g^2)$
$p = 2$, g *even*	$14g^2 + 6g - 6 + \frac{1}{q}O(g^3)$	$\frac{3}{2}g + 2 + \frac{1}{q}O(g^2)$
$p = 2$, g *odd*	$14g^2 + 7g - 3 + \frac{1}{q}O(g^3)$	$\frac{3}{2}g + \frac{5}{2} + \frac{1}{q}O(g^2)$

On average, a divisor can be doubled with the following numbers of field operations:

	multiplications	*inversions*
$p \neq 2$, g *even*	$16g^2 + 7g - 6 + \frac{1}{q}O(g^3)$	$\frac{3}{2}g + 2 + \frac{1}{q}O(g^2)$
$p \neq 2$, g *odd*	$16g^2 + 8g - 3 + \frac{1}{q}O(g^3)$	$\frac{3}{2}g + \frac{5}{2} + \frac{1}{q}O(g^2)$
$p = 2$, $h = 1$, g *even*	$7g^2 + 3g - 3 + \frac{1}{q}O(g^3)$	$\frac{1}{2}g + 2 + \frac{1}{q}O(g^2)$
$p = 2$, $h = 1$, g *odd*	$7g^2 + 4g + \frac{1}{q}O(g^3)$	$\frac{1}{2}g + \frac{5}{2} + \frac{1}{q}O(g^2)$
$p = 2$, $h = X$, g *even*	$11g^2 + 4g - 3 + \frac{1}{q}O(g^3)$	$\frac{1}{2}g + 3 + \frac{1}{q}O(g^2)$
$p = 2$, $h = X$, g *odd*	$11g^2 + 5g + \frac{1}{q}O(g^3)$	$\frac{1}{2}g + \frac{7}{2} + \frac{1}{q}O(g^2)$

Conclusion 15 *For large q, the arithmetic in hyperelliptic Jacobians of genus at least 2 is faster in even characteristic with $h \in \{1, X\}$ than in odd characteristic. The number of field multiplications and inversions strictly increases with the genus.*

3.3 The average bit complexity

While Theorem 14 shows that the number of field operations increases strictly as the genus grows, the cryptographic setting described in Section 1.4 implies that at the same time the field size decreases because $\log q = \frac{c}{g}$ for some constant security parameter c. Hence the field operations may become more efficient, and the bit complexity of adding divisors depends very much on the implementation of the field arithmetic. We examine three basic situations in which the complexity of field operations is constant or grows with $\log q$ or $\log^2 q$.

Throughout this section we assume that c is "sufficiently large" and g "not too large" compared to c, so that the terms "$\frac{1}{q}O(g^4)$" etc. in the complexities can be neglected. As justified in Section 1.4, this condition is fulfilled in practice. To assure mathematical rigour, we may assume that $g \leq c^{\varepsilon}$ for some constant $\varepsilon < 1$. Then

$$\frac{1}{q}O(g^4) = \frac{g^4}{e^{c/g}}O(1) \subseteq \frac{c^{4\varepsilon}}{(e^{1-\varepsilon})^c}O(1) \subseteq o(1) \text{ for } c \to \infty,$$

and the following discussion yields asymptotic results for $c \to \infty$.

If the finite field is so small that its elements can be represented in single registers and the field operations can be performed with single precision, then they take constant time, and the assertion of Theorem 14 directly carries over to the bit complexity.

In custom-designed hardware for finite field arithmetic the complexity of the field operations usually grows with $\log q$. Thus, assume that a multiplication needs time $m \log q$ and an inversion $i \log q$. As we get technical problems when comparing the arithmetics for genus g and $g+1$, one being even and one odd, and a too fine-grained analysis is not even meaningful in our rather coarse model, we restrict our comparison to either odd or even genus, i.e. remain within one row of the tables in Theorem 14. Then, an operation in the Jacobian can be executed in an average time of

$$(\alpha g^2 + \beta g - \gamma)m \log q + (\delta g + \eta)i \log q = c\left(\left(\alpha g + \beta - \frac{\gamma}{g}\right)m + \left(\delta + \frac{\eta}{g}\right)i\right)$$

for some non-negative constants α, β, γ, δ, η. The derivative of this expression with respect to g is positive whenever

$$\frac{i}{m} < \frac{4\alpha + \gamma}{\eta} \leq \frac{\alpha g^2 + \gamma}{\eta}.$$

Plugging in the actual values of Theorem 14 shows that this is the case for $i/m \leq 11$, which is fulfilled in practice (cf. [Ber68], Chapters 2.3 and 2.4). Hence in this situation the arithmetic in hyperelliptic Jacobians becomes slower as the genus grows.

In a generic software implementation, the field operations usually take time proportional to $\log^2 q$, say a multiplication needs $m \log^2 q$ and an inversion $i \log^2 q$. Then the operations in the Jacobians take time

$$c^2 m \left(\alpha + \frac{\beta g - \gamma}{g^2}\right)$$

for the field multiplications and

$$c^2 i \frac{\delta g + \eta}{g^2}$$

for the inversions. Thus the effort for field inversions is strictly decreasing. Also, with the constants set forth in Theorem 14, the effort for multiplications is strictly decreasing except for the step between $g = 2$ and $g = 4$ in odd characteristic. This conclusion, however, has to be approached with care, as it is based on the non-leading coefficients β and γ, which are bound to change already when two multiplications are interchanged. Anyway, the relative speed-up obtainable from the multiplications is very limited. Considering, for instance, the doubling step for $p = 2$, $h = 1$ and g even, we observe that the total effort for multiplications varies between $7.75mc^2$ (for $g = 2$) and $7mc^2$ (for $g \to \infty$). The effect is even less pronounced in the other cases with a higher value of α.

How advantageous it is with respect to the inversions to increase the genus depends on the ratio i/m. The larger this ratio is, the more one can profit from a higher genus. In any case, the hyperbolic shape of the complexity curve results in fast dropping marginal profits.

3.4 Concluding remarks

The discussion of the preceeding section shows that given a desired security level, i.e. an appropriate group size, hyperelliptic cryptosystems should preferably be implemented in characteristic 2. If the complexity of the field operations is constant or grows with $\log q$, then the smallest possible genus fitting the system requirements should be chosen. If the complexity of the field operations grows with $\log^2 q$, a higher genus, to be determined by computational experiments, might be recommendable. For instance, the smallest genus such that the field elements fit into one machine word could be a good choice; from then on, the efficiency would drop again.

In any case, due to optimised formulae available for adding points on elliptic curves, hyperelliptic arithmetic based on the generic algorithms of this article is less efficient than elliptic curve arithmetic. It might be worthwhile to investigate explicit formulae for the addition law in low-genus hyperelliptic Jacobians. Spallek, for instance, reports such an implementation of genus 2 curves which is slower than elliptic curve arithmetic by a factor of only 2.5 for $q^g \approx 10^{40}$ ([Spa94], p. 31). Further experimental results can be found in [Kri97].

Acknowledgements

I thank Alfred Menezes, Nigel Smart and Andreas Stein for fruitful discussions and, in particular, for pointing out the relevance of the reduction step to me. I am grateful to my supervisor, Dieter Jungnickel, for helpful suggestions concerning the final revision of this paper. Part of this article was written while I was visiting the Department of Combinatorics and Optimization of the University of Waterloo, Canada. I am grateful to my colleagues for their hospitality and especially to Scott Vanstone, whose financial support made my stay possible and who initiated this study.

References

[ADH94] Leonard M. Adleman, Jonathan DeMarrais, and Ming-Deh Huang. A subexponential algorithm for discrete logarithms over the rational subgroup of the Jacobians of large genus hyperelliptic curves over finite fields. In *[AH94]*, pages 28–40, 1994.

[AH94] Leonard M. Adleman and Ming-Deh Huang, editors. *Algorithmic Number Theory*, volume 877 of *Lecture Notes in Computer Science*, Berlin, 1994. Springer-Verlag.

[Art24] E. Artin. Quadratische Körper im Gebiete der höheren Kongruenzen I. *Mathematische Zeitschrift*, 19:153–206, 1924.

[Ber68] Elwyn R. Berlekamp. *Algebraic Coding Theory.* McGraw-Hill Series in Systems Science. McGraw-Hill, New York, 1968.

[Buh98] J. P. Buhler, editor. *Algorithmic Number Theory — ANTS-III*, volume 1422 of *Lecture Notes in Computer Science*, Berlin, 1998. Springer-Verlag.

[Can87] David G. Cantor. Computing in the Jacobian of a hyperelliptic curve. *Mathematics of Computation*, 48(177):95–101, 1987.

[Eng99a] Andreas Enge. Computing discrete logarithms in high-genus hyperelliptic Jacobians in provably subexponential time. Combinatorics and Optimization Research Report CORR 99-04, University of Waterloo, February 1999. Available at http://-cacr.math.uwaterloo.ca/techreports/1999/corr99-04.ps; to appear in *Mathematics of Computation*.

[Eng99b] Andreas Enge. *Elliptic Curves and Their Applications to Cryptography — An Introduction*. Kluwer Academic Publishers, 1999.

[ES00] Andreas Enge and Andreas Stein. Smooth ideals in hyperelliptic function fields. Combinatorics and Optimization Research Report CORR 2000-08, University of Waterloo, January 2000. Available at http://cacr.math.uwaterloo.ca/techreports/2000/corr2000-08.ps.

[FR94] Gerhard Frey and Hans-Georg Rück. A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves. *Mathematics of Computation*, 62(206):865–874, April 1994.

[Gau01] Carl Friedrich Gauß. *Disquisitiones Arithmeticae*. Gerh. Fleischer Jun., Leipzig, 1801.

[Gra73] De la Grange. Recherches d'arithmétique. *Nouveaux Mémoires de l'Académie Royale des Sciences et Belles-Lettres*, pages 265–312, 1873.

[KK88] Arnold Knopfmacher and John Knopfmacher. The exact length of the Euclidian algorithm. *Mathematika*, 35:297–304, 1988.

[Knu81] Donald Ervin Knuth. *The Art of Computer Programming*, volume 2 - Seminumerical Algorithms. Addison-Wesley, Reading (Massachusetts), 2nd edition, 1981.

[Kob89] Neal Koblitz. Hyperelliptic cryptosystems. *Journal of Cryptology*, 1:139–150, 1989.

[Kob98] Neal Koblitz. *Algebraic Aspects of Cryptography*, volume 3 of *Algorithms and Computations in Mathematics*. Springer-Verlag, Berlin, 1998.

[Kri97] Uwe Krieger. signature.c — Anwendung hyperelliptischer Kurven in der Kryptographie. Master's thesis, Universität Essen, Deutschland, 1997.

[Lew71] Donald Lewis, editor. *Proceedings of Symposia in Pure Mathematics*, volume 10, Providence (Rhode Island), 1971. American Mathematical Society.

[Ma87] Keju Ma. Analysis of polynomial gcd computations over finite fields. Master's thesis, University of Toronto, Toronto, 1987.

[McC90] Kevin S. McCurley. The discrete logarithm problem. In *[Pom90]*, pages 49–74, 1990.

[MG90] Keju Ma and Joachim von zur Gathen. Analysis of Euclidian algorithms for polynomials over finite fields. *Journal of Symbolic Computation*, 9:429–455, 1990.

[MOV93] Alfred J. Menezes, Tatsuaki Okamoto, and Scott A. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Transactions on Information Theory*, 39(5):1639–1646, September 1993.

[MST99] Volker Müller, Andreas Stein, and Christoph Thiel. Computing discrete logarithms in real quadratic congruence function fields of large genus. *Mathematics of Computation*, 68(226):807–822, 1999.

[MWZ98] Alfred J. Menezes, Yi-Hong Wu, and Robert J. Zuccherato. An elementary introduction to hyperelliptic curves. In *[Kob98]*, pages 155–178. Springer-Verlag, 1998.

[Pom90] Carl Pomerance, editor. *Cryptology and Computational Number Theory*, volume 42 of *Proceedings of Symposia in Applied Mathematics*, Providence (Rhode Island), 1990. American Mathematical Society.

[PS98] Sachar Paulus and Andreas Stein. Comparing real and imaginary arithmetics for divisor class groups of hyperelliptic curves. In *[Buh98]*, pages 576–591, 1998.

[SA98] Takakazu Satoh and Kiyomichi Araki. Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves. *Commentarii Mathematici Universitatis Sancti Pauli*, 47(1):81–92, 1998.

[Sem98] I. A. Semaev. Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p. *Mathematics of Computation*, 67(221):353–356, 1998.

[Sha71] D. Shanks. Class number, a theory of factorization and genera. In *[Lew71]*, pages 415–440, 1971.

[Sil86] Joseph H. Silverman. *The Arithmetic of Elliptic Curves*. Graduate Texts in Mathematics. Springer-Verlag, New York, 1986.

[Sma99] Nigel P. Smart. The discrete logarithm problem on elliptic curves of trace one. *Journal of Cryptology*, 12(3):193–196, 1999.

[Spa94] Anne-Monika Spallek. *Kurven vom Geschlecht 2 und ihre Anwendung in Public-Key-Kryptosystemen*. PhD thesis, Universität Gesamthochschule Essen, 1994.