# The point counting problem for curves over finite fields

*Graeme Taylor*

First-Year Report
School of Mathematics
University of Edinburgh
May 2007

# Abstract

The group law on elliptic curves is well-known and gives rise to elliptic curve cryptography systems which find application to government and industry today. However, the generalisation to higher genus requires the manipulation of divisor classes rather than points, and analogues of key genus 1 results have yet to be found. Nonetheless, effective computation within the group is possible, and techniques for finding the cardinality of hyperelliptic curve jacobians are improving, with calculations over curves of cryptographically significant size having recently been achieved. This report sets out the mathematical background to this problem, its cryptographic application and a solution strategy from algebraic geometry; and discusses the two main approaches employed in its attack.

# Contents

# Program of Studies

## Themes

I began my studies with [10], the standard reference for elliptic curves. I developed *Maple* procecures for explicit computation of the group law, and became particularly interested in curves over finite fields. I also spent some time looking at ideas connected to the height of points on elliptic curves, such as szpiro ratios, the abc conjecture and elliptic divisibility sequences, again using Maple.

These lead me to the broader theme of rational points of curves over finite fields. Initially I looked at the Weil theorems and the connection between the number-theoretic properties of zeta functions and the cardinality of corresponding curves, in the context of controlling the 'defect' (that is, seeking curves with large numbers of points). Again, this had a highly computational flavour.

I then became interested in the technical challenges presented by computation of this cardinality for elliptic curves. I started to use the Computer Algebra System SAGE ([S]), and studied the SEA algorithm and related material, such as modular polynomials, specific to elliptic curves.

The absence of equivalent techniques for higher genus, and the approaches employed as a result, have been the focus of my attention since, and are the subject of this report. Following Cantor ([1],[2]), Gaudry ([4],[5]) and several sections of [3] I have written Maple and SAGE/Python 'black-box' programs for explicit computation with rational divisor classes of hyperelliptic curves, as well as implementing generic algorithms for tasks such as integer multiplication or order determination. I am particularly interested in the interaction between such techniques and modular approaches.

## Summary of Activities

I maintain a research weblog ([W]) through which summary notes and source code on many of the above topics are available.

During the year I attended Dr. Cheltsov's *Complex Algebraic Surfaces* (semester 1) and LFCS taught postgraduate courses on *Gröbner Bases* (semester 1) and *Formal analysis of cryptographic protocols* (semester 2).

*The Geometry Club* provides a forum for postgraduates to present seminars on their work; I attended a few of these, and gave one on *Hyperelliptic Curves over finite fields* in semester 2.

# Chapter 1

# Introduction

## 1.1 Preliminaries

**Definition 1.** Let $f \in K[u]$ be a squarefree monic polynomial of degree $2g+1$ and $h \in K[u]$ be of degree at most $g$. Then a curve with affine model

$$C : v^2 - h(u)v = f(u)$$

is described as a *hyperelliptic curve of genus g*. The special case of genus 1, gives an *elliptic curve*.

Throughout, we assume that $K = \mathbb{F}_q$ is a finite field of characteristic not 2 or $2g+1$, with algebraic closure $\bar{K}$.

**Definition 2.** A pair $P = (x, y) \in \bar{K} \times \bar{K}$ is described as a *point of C* if $y^2 - h(x)y = f(x)$. The point is *rational* if $(x, y) \in K \times K$.
Since we work in affine rather than projective space, there is also the *rational point at infinity*, $\infty$.

The well-known "chord and tangent" procedure equips the set of rational points of an elliptic curve with a group structure, and thus determining the number of such points is equivalent to finding the cardinality of the group. However, for curves of higher genus there is no group structure on the points, which gives rise to two possible generalisations of the *point counting problem*:

- For a curve $C$, compute $\#X(K)$, the number of rational points over $K$.
- For a curve $C$, construct a finite group $G$ containing the rational points (thus equipping them with a group law) and compute its cardinality.

In fact, through an appropriate choice of the group $G$, these problems are connected. To achieve this, we introduce divisors and the Jacobian of a curve as follows:

## 1.2 Divisors

**Definition 3.** A *divisor* $D$ of $C$ is a finite formal sum of points of $C$:

$$D = \sum_i{}' m_i P_i \;\; m_i \in \mathbb{Z}$$

Its *degree* is given by $\sum_i m_i$.

**Definition 4.** The *group of divisors* $\mathrm{Div}_C$ is the set of divisors equipped with formal (pointwise) addition; it has a subgroup, $\mathrm{Div}_C^0$, consisting of the degree 0 divisors.

Any polynomial $p(u, v)$ can be considered as a function on $C$ of the form $p = a(u) + b(u)v$, since $v^2 = f(u)$.
If $p$ vanishes at $(x, y)$ then the order of the zero $(x, y)$ of $p$ is the exponent of the highest power of $(u - x)$ which divides $a - b^2 f$.

**Definition 5.** Thus we can denote functions from $K(C)$ as $h = p/q$ for $p, q \in K[u, v]$ such that $v^2 - f \not| q$: that is, $q$ is not everywhere zero on $C$. Then $h$ will have a finite set of zeros (those of $p$) and of poles (zeros of $q$); we associate to $h$ a divisor, $(h)$, where the $P_i$ are those zeros and poles and $m_i$ their multiplicities:

$$(h) := \mathrm{div(h)}_0 - \mathrm{div(h)}_\infty = \sum_{\substack{P_i \in \\ \{zeros\, of\, p\}}} ord_{P_i}(p) P_i - \sum_{\substack{P_i \in \\ \{zeros\, of\, q\}}} ord_{P_i}(q) P_i$$

**Definition 6.** If there is a nonzero function $h$ on $C$ such that $D$ a divisor is $(h)$, then $D$ is described as *principal*.

The set $\mathrm{Princ}_C$ of principal divisors is a subgroup of $\mathrm{Div}_C^0$.

**Definition 7.** The *divisor class group of $C$ of degree zero* or *Picard group of $C$*, (equivalently here the *Jacobian of $C$*) is the quotient group

$$\mathrm{Pic}_C^0 = \mathrm{Div}_C^0 / \mathrm{Princ}_C$$

Thus $D_1, D_2$ are in the same class if $\exists f \in K(C)$ s.t. $\mathrm{div}(f) = D_1 - D_2$.

Any divisor $D \in \mathrm{Div}_C^0$ will have a representative of *weight* $r$

$$D = \sum_{i=1}^{r} P_i - r\infty$$

such that if $P_i$ is a point in the sum, then $P_j \neq -P_i$ for any $j \neq i$. Such a representation is called *semi-reduced*.
If $r \leq g$ the representation is called *reduced*. By Riemann-Roch, any divisor class in $\mathrm{Pic}_C^0$ has a reduced representative.

## 1.3 Mumford polynomials, rationality and the group law

**Definition 8.** Let $D$ be a semi-reduced divisor whose points are $P_i = (x_i, y_i)$. We associate to $D$ polynomials $a, b \in \bar{K}[u]$ such that

$$a(u) = \prod_i^r (u - x_i)$$

$$b(x_i) = y_i \; 1 \leq i \leq r$$

such that $b$ has degree less than that of $a$, and the appropriate multiplicity for repeated points- i.e., if $P_i$ occurs $k$ times in the semi-reduced representation of $D$, then $(u - x_i)^k$ divides $b - y_i$. We write

$$D = div(a, b)$$

**Definition 9.** $D = div(a, b)$ is *rational* if $a, b \in K[u]$.

Thus a rational divisor needn't be the sum of rational points! However, this will be true of a weight 1 rational divisor; which is a single rational point thought of as a divisor. For an elliptic curve, therefore, the existence of weight 1 representatives for sums of rational points ensures that the group of rational divisors *is* isomorphic to the rational points of the curve. As previously mentioned, this fails in higher genus: we can now see that this is because the sum of two points as a weight 2 divisor needn't reduce further.

However, the set of rational divisors $\mathcal{J}_K(C)$ *is* closed under the group operation of $\mathrm{Pic}_C^0$ (inherited from $\mathrm{Div}_C^0$) and each rational point of $C$ corresponds to a distinct divisor class. Thus $\mathcal{J}_K(C)$ is a suitable generalisation to higher genus of the group of rational points of an elliptic curve. In the next section an explicit description of the group law will be given in terms of the Mumford polynomial representation, but a sketch of the procedure can already be given:

- Given divisor classes $\mathcal{D}_1, \mathcal{D}_2 \in \mathcal{J}_K(C)$, take rational reduced representatives $D_1, D_2$.
- Form a new rational, semi-reduced divisor $D_1 + D_2$ by combining the points of $D_1, D_2$ in $\mathrm{Div}_C^0$.
- Reduce modulo $\mathrm{Princ}_C$ to some rational $D$ of degree at most $g$.
- Define $\mathcal{D}_1 \oplus \mathcal{D}_2$ to be the equivalence class of $D$ in $\mathcal{J}_K(C)$.

## 1.4 Applications

Both the set of rational points $X(K)$ and the group of rational divisors $\mathcal{J}_K(C)$ have found application in opposite ends of the communication industry: error-

correcting codes (ensuring the readability of messages) and cryptography (restricting the readability of messages). For a discussion of the connection between algebraic curves and codes, see [6]. The use of hyperelliptic curves in cryptography will be discussed here since it depends on techniques also of relevance to the point counting problem.

## The Discrete Logarithm Problem

In the late 1970s asymmetric key encryption was introduced: this depends upon "one-way functions": functions which are easy to evaluate but computationally infeasible to invert without an extra piece of information. Thus one can secretly choose a "private key", and safely disclose the encryption procedure in the form of a "public key"; messages encrypted by the public key can only be read by the holder of the private key, which need never be shared. Asymmetric key systems tend to be slower than traditional symmetric encryption of equivalent strength, but those require prior arrangement of a secret key: an unreasonable expectation in scenarios such as e-commerce. The two are thus typically combined by the use of asymmetric encryption to securely exchange a symmetric key, but the distinction of public/private keys also allows for advances such as digital signatures.

Whilst no true one-way functions have been proven to exist, there are two strong candidates: the original approach of RSA (based on the difficulty of factorisation) and the more recent discrete logarithm problem for finite groups. The NSA currently recommends elliptic-curve cryptography over RSA for public key systems due to its greater security per bit: for instance, securing a 128-bit symmetric key requires a 3072-bit RSA key, but only a 256-bit elliptic curve key; for a 256-bit symmetric key, these grow to 15360 and 521 bits respectively. As will be illustrated later, curves of higher genus offer larger group sizes relative to that of the underlying finite field; the first genus 2 curves of cryptographic strength were demonstrated in 2004 by Gaudry and Schost (see [5]).

Let $(G, \oplus)$ be an additive cyclic group of prime order $p$ generated by an element $g$. We can define a map

$$\varphi : \mathbb{Z} \to G$$

$$n \mapsto [n]g = \underbrace{g \oplus g \oplus \cdots \oplus g}_{n \text{ copies}}$$

This gives an isomorphism between $(\mathbb{Z}/p\mathbb{Z}, +)$ and $(G, \oplus)$. Given $g, h \in G$, the *Discrete Logarithm Problem (DLP)* is to find $k \in \mathbb{Z}$ such that $[k]g = h$.

The difficulty of the DLP depends in a vital way upon the underlying group. For the obvious choice of a group isomorphic to $(\mathbb{Z}/p\mathbb{Z}, +)$, namely itself, recovering the discrete logarithm is not difficult:

**Example 1.** Let $g = a + p\mathbb{Z}$ be a generator of $(\mathbb{Z}/p\mathbb{Z}, +)$, and $h = b + p\mathbb{Z}$ another element. Then the DLP

$$[k]g = h$$

has solution

$$k = a^{-1}b \pmod{p}$$

and this calculation is of polynomial complexity in $p$.

For secure DLP cryptography we need a cyclic group such that computation of the group law is efficient, but the isomorphism with $\mathbb{Z}/p\mathbb{Z}$ is not apparent from the group elements. It is believed that the groups of rational points on elliptic curves / rational divisors on hyperelliptic curves are indeed suitable.

# Chapter 2

# Explicit computation in the Jacobian group

In [1] the first explicit formulae for composition of divisor classes are given. These are carefully constructed to avoid decomposing a divisor into its constituent points (which even for a rational divisor may be in $\bar{K}\backslash K$) by consideration of gcds and resultants to identify the problem cases of common points between the divisors. A reduction procedure is also given, which depends on the observation that $D = div(a, b)$ is equivalent to $E = -((b - v) - D)$ which is of lesser weight. Instead of Cantor's original algorithm we present here the version from [7], which corresponds to the more general model of definition 1:

**Theorem 1.** *Explicit Group law equations*
***Composition***
*INPUT:* $D_1 = div(u_1, v_1), D_2 = div(u_2, v_2)$; $C : y^2 + h(x)y = f(x)$
*OUTPUT:* $D = div(u, v)$ *semi-reduced such that* $D = D_1 + D_2$ .

1. *compute* $d_1 = gcd(u_1, u_2) = e_1u_1 + e_2u_2$
2. *compute* $d = gcd(d_1, v_1 + v_2 + h) = c_1d_1 + c_2(v_1 + v_2 + h)$
3. *set* $s_1 = c_1e_1, s_2 = c_1e_2, s_3 = c_2$ *such that* $d = s_1u_1 + s_2u_2 + s_3(v_1 + v_2 + h)$
4. $u = \dfrac{u_1u_2}{d^2}$
5. $v = \dfrac{s_1u_1v_2 + s_2u_2v_1 + s3(v_1v_2 + f)}{d}$ *mod* $u$

***Reduction***
*INPUT:* $D = div(u, v)$ *semi-reduced.*
*OUTPUT:* $D' = div(u', v')$ *reduced such that* $D \equiv D'$ .

1. *set* $u' = \dfrac{f - vh - v^2}{u}$, $v' = (-h - v)$ *mod* $u'$
2. *If* $deg(u') > g$, *set* $u = u', v = v'$
   *goto step 1*
3. *make* $u'$ *monic*

No freely-available computer algebra system currently supports working in the Jacobian of a hyperelliptic curve over finite fields. SAGE ([S]) is capable of such computations over $\mathbb{Q}$, and using the above I have extended its procedures to work with finite fields. I also improved the multiplication-by-n calculations (to use $\log_2(n)$ instead of $n$ group operations) and introduced order-finding via Terr's Baby Step, Giant Step variant (described in [3]). These modifications (see appendix) are currently under peer review for potential inclusion in the main SAGE distribution, and can be obtained from my website. I have also released an implementation in Maple for prime fields.

# Chapter 3

# Characteristic Polynomial of Frobenius

An alternative characterisation of rational divisors can be obtained via the Frobenius endomorphism:

**Definition 10.** The *Frobenius morphism* is the map $\phi_q : \alpha \mapsto \alpha^q$. It extends naturally to points of $\bar{K}$; to polynomials over $\bar{K}$ coefficient-wise, and hence to divisors $div(a, b)$, leading to $\pi$, the *Frobenius endomorphism* of $\mathrm{Pic}^0_C$.

A polynomial is fixed by $\pi$ if and only if its coefficients are from $K$; hence $\mathcal{J}_K(C) = ker(id_{\mathrm{Pic}^0_C} - \pi)$ and so

$$\#\mathcal{J}_K(C) = deg(id_{\mathrm{Pic}^0_C} - \pi)$$

$\pi$ acts linearly as an element of $\mathrm{End}(\mathrm{Pic}^0_C)$: denoting its characteristic polynomial as $\chi(T)$, we have

$$\#\mathcal{J}_K(C) = \chi(1)$$

**Theorem 2.** *(Weil Theorems)*
*$\chi(T)$ is a monic integer polynomial of degree $2g$ with roots $\lambda_i$ (the eigenvalues of Frobenius) of absolute value $\sqrt{q}$.*

This immediately gives us a bound for the cardinality of the group, which will motivate a number of search strategies.

**Corollary 3.** *(Weil Interval)*

$$(\sqrt{q} - 1)^{2g} \leq \#\mathcal{J}_K(C) \leq (\sqrt{q} + 1)^{2g}$$

**Example 2.** For a hyperelliptic curve of genus 2, we have:

- $\chi(T) = T^4 - s_1T^3 + s_2T^2 - qs_1T + q^2$
- $\#\mathcal{J}_K(C) = 1 - s_1 + s_2 - qs_1 + q^2 \leq (\sqrt{q} + 1)^4$
- $|s_1| \leq 4\sqrt{q}$, $|s_2| \leq 6q$.

# Chapter 4

# Search strategies

## 4.1 Square-root algorithms

From the results of the previous chapter, it suffices to compute $\mathcal{J}_K(C)$ modulo $w = (\sqrt{q}+1)^4 - (\sqrt{q}-1)^4 = 2\lfloor 4(q+1)\sqrt{q} \rfloor$. This motivates modular approaches (see the following section), but purely group-theoretic techniques can also be applied.

**Theorem 4.** *Generic group order algorithm*
*INPUT: A group $G$ and interval $[a,b]$ of width $w$ s.t. $|G| \in [a,b]$.*
*OUTPUT: The order of $G$.*

1. *Generate a random element $g \in G$.*
2. *compute n=ord(g) and set $e = n$.*
3. *While $e < w$:*
   *Generate a random element $g \in G$.*
   *set $e = lcm\{e, ord(g)\}$*
4. *Return the unique $N \in [a,b]$ s.t. $N \equiv 0$ mod $e$.*

The quantity $e$ converges to the exponent of the group, which necessarily divides its order N (so $N \equiv 0$ modulo $e$) . However, the algorithm will therefore fail to terminate for groups where the exponent is less than $w$.

Moreover, although the algorithm requires only a "black box" for the group law (developed for hyperelliptic curves in chapter 2) and random element generation (easy for $\mathcal{J}_K(C)$ by generating random rational points of $C$), it depends upon the calculation of the order of such elements. This can be seen as a special case of the discrete logarithm problem, the computational difficulty of which is the reason for interest in hyperelliptic curve cryptography!

The best generic algorithms for the discrete logarithm problem of recovering $n$ such that $[n]g = h$ for some $g, h \in G$ - the *Baby Steps Giant Steps algorithm* and *Pollard's Rho method* - have complexity of order $\sqrt{n}$, and are thus known as square-root algorithms. It was conjectured that a similar complexity bound would hold for order calculation, but a very recent result [11] shows that a lower bound holds there, and in special cases significantly faster computation may be

possible. I am looking in to how these techniques could be optimised for groups of rational divisors.

## 4.2 Schoof's Algorithm

Since the cardinality of $\mathcal{J}_K(C)$ and the coefficients of $\chi(T)$ are bounded, their exact values can be recovered from their values modulo a selection of primes via the chinese remainder theorem. For $l$ coprime to $p$, the characteristic polynomial of $\pi$ modulo $l$ is the characteristic polynomial of $\pi$ restricted to the $l$-torsion subgroup.

### 4.2.1 Elliptic Curves

In the genus 1 case, the $l$-torsion elements are characterised by the division polynomials $\phi_l$, which can be computed recursively. Further, $\chi$ takes a particularly simple form, such that we need only determine its *trace* $t$ satisfying

$$\pi^2 - [t]\pi + [q] = [0]$$

This leads to Schoof's algorithm:

**Theorem 5.** ***Schoof's algorithm in genus 1***
*INPUT: Curve $E/\mathbb{F}_q$*
*OUTPUT: $\#E(\mathbb{F}_q)$ the cardinality of $E$.*

1. *Compute $L$ a set of primes such that*

$$\prod_{l \in L} l \geq 4\sqrt{q} \tag{4.1}$$

   *with $L$ minimal*

2. *For each $l \in L$, compute $t_l$, the trace modulo $l$.*
3. *By the Chinese Remainder theorem, find $t_L$, the trace modulo $\prod_{l \in L} l$.*
4. *Expressing $t_L$ as $t$ in the range $-2\sqrt{q} \leq t \leq 2\sqrt{q}$ gives the true trace.*
5. *Return $q + 1 - t$.*

Schoof's original approach to step 2 is to recover $t_l$ by brute force determination of a $\tau \in 1, \ldots l-1$ such that

$$(x^{q^2}, y^{q^2}) \oplus [q_l](x, y) = [\tau](x^q, y^q)$$

There are explicit formulae for the multiplication-by-$m$ isogeny, and intermediate expressions are controlled by working modulo the $l$-division ideal generated by $\phi_l$ (with coefficients further constrained modulo $l$). This algorithm is then of

polynomial time complexity, but is still too slow for curve sizes of cryptographic interest: the $\phi_l$ are of degree $(l^2-1)/2$ and hence impractical beyond $q \approx 10^{200}$.

Refinements by Elkies and Atkin give rise to the SEA algorithm: by way of modular polynomials, primes $l$ can be characterised as either Elkies or Atkin type. For the former, $\phi_l$ can be replaced by a factor of degree $(l-1)/2$, whilst the latter give rise to an easily computed set of candidates for $t_l$ and hence $t$ which can be tested against random points, although the size of this set grows exponentially and thus careful choice of primes in stage 1 is necessary. Without precomputation, the determination of the modular polynomials can be harder than naive determination of $t_l$. Nonetheless, the SEA algorithm introduces significant performance gains and is effective for $q \approx 10^{500}$: in November 2006 a record computation over $\mathbb{F}_p$ with $p = 10^{2499} + 7131$ was completed (although this took over a year of computation).

### 4.2.2 Hyperelliptic Curves

However, the SEA algorithm does not readily lift to higher genus: one has to work with $l$-division ideals which may not be principal (unlike the elliptic case), and although hyperelliptic analogues of the modular polynomials have been developed, they have not lead to an Elkies procedure.

In [4] and [5], the authors develop a series of polynomial constraints on $l$-torsion divisors of curves of genus 2: for a reduced divisor $\mathcal{D} = P_1 + P_2$ to satisfy $[l]\mathcal{D} = 0$ we obviously require $[l]P_1 = -[l]P_2$. Using explicit formulae for the multiplication-by-$l$ (from Cantor [2]) This gives a series of four polynomial equations for the four unknowns $x_1, y_1 = P_1$, $x_2, y_2 = P_2$. However, these coefficients needn't be in $K$.

Given the ability to construct such divisors, there is then a hyperelliptic equivalent for Schoof's original algorithm:

**Theorem 6.** ***Schoof's algorithm in genus 2***
*INPUT: Curve $C/\mathbb{F}_q$*
*OUTPUT: $\#\mathcal{J}_K(C)$.*

1. *For sufficiently many small primes $l$:*
   *Set $L = \{(s_1, s_2); s_1, s_2 \in [0, l-1]\}$.*
   *While $\#L > 1$ do:*
   - *Construct an $l$-torsion divisor $D$*
   - *Eliminate elements of $L$ such that*

     $$\pi^4(D) - s_1\pi^3(D) + s_2\pi^2(D) - (qs_1 \bmod l)\pi(D) + (q^2 \bmod l)D \neq 0$$

   - *Deduce $\chi(T) \bmod l$ from the final pair $s_1, s_2$.*
2. *Construct $\chi(T)$ from the $\chi(T) \bmod l$ by the chinese remainder theorem.*
3. *Return $\chi(1)$*

Note ([4]) that $\#X(C)$ can also be recovered as $q - s_1$.

## 4.3 Hybrid Algorithms

The genus 2 approach described above is only feasible for $l$ at most 13. Various other techniques introduce further modular information. By a halving algorithm (see [4]), divisors of order $2^k$ for increasing $k$, and hence $\chi(T)$ mod $2^e$ for some $e$, can be constructed. A technique based on the Cartier-manin operator can recover $\chi(T)$ mod the characteristic $p$ of the underlying field, although this costs time exponential in $\log p$.

This modular information can then be incorporated into a BSGS algorithm, speeding the computation by ignoring incompatible steps. The algorithm outline below is given in [8], and as implemented in [5] allows for cardinalities of around 164-bit size to be computed in about a week.

**Theorem 7.** ***Gaudry-Harley point counting algorithm***
*INPUT: Curve $C/\mathbb{F}_q$ of genus 2.*
*OUTPUT: $\#\mathcal{J}_K(C)$.*

1. *Compute $\#\mathcal{J}_K(C)$ mod $2^e$ by the halving algorithm.*
2. *For primes $l = 2, 3, 5, \ldots, l_{max}$:*
   - *Compute $\chi(T)$ mod $l$ by a Schoof-like algorithm.*
   - *Compute $\#\mathcal{J}_K(C)$ mod $l$ from $\chi(T)$ mod $l$.*
3. *Compute $\chi(T)$ mod $p$ via the Cartier-Manin operator.*
4. *Compute $\#\mathcal{J}_K(C)$ mod $p$ from $\chi(T)$ mod $p$.*
5. *Compute $\#\mathcal{J}_K(C)$ mod $m = 2^e \cdot 3 \cdots l_{max} \cdot p$ by CRT.*
6. *Compute $\#\mathcal{J}_K(C)$ by a square root algorithm that exploits knowledge of $\#\mathcal{J}_K(C)$ mod $m$.*

Most steps of the above algorithm are subject to feasibility constraints and have potential for refinement or replacement; further, the advances in generic group order computation may provide practical means of recovering additional information about $\#\mathcal{J}_K(C)$ that would supplement this modular approach.

# Bibliography


[1] D.G. Cantor 'Computing in the Jacobian of an hyperelliptic curve' *Math. Comp., 48(177):95-101, 1987*

[2] D.G. Cantor 'On the analogue of the division polynomials for hyperelliptic curves' *J. Reine Angew. Math., 447:91-145, 1994*

[3] H. Cohen, G. Frey et al. 'Handbook of Elliptic and Hyperelliptic Curve Cryptography' *Chapman and Hall/CRC 2006*

[4] P. Gaudry, R. Harley 'Counting points on hyperelliptic curves over finite fields.' *ANTS-IV (2000) W.Bosma, Ed. vol. 1838 of Lecture notes in Comput. Sci., Springer-Verlag, pp. 297-312.*

[5] P. Gaudry, and É. Schost 'Construction of secure random curves of genus 2 over prime fields'. *Advances in Cryptology - EUROCRYPT 2004 (2004), C. Cachin and J. Camenisch, Eds., vol. 3027 of Lecture Notes in Comput. Sci., Springer-Verlag, pp. 239-256.*

[6] G. van der Geer 'Curves over Finite Fields and Codes' *Proceedings of 3ECM, Barcelona, 2000.*

[7] T.Lange 'Formulae for arithmetic on genus 2 hyperelliptic curves' To appear in *Journal of Applicable Algebra in Engineering, Communication and Computing.*

[8] K. Matsuo, J.Chao and S. Tsujii 'An improved Baby Step Giant Step Algorithm for Point Counting of Hyperelliptic Curves over Finite Fields' *ANTS-V vol. 2369 of Lecture Notes in Comput. Sci., Springer-Verlag pp. 461-474.*

[9] "The Case for Elliptic Curve Cryptography" NSA Central Security Service http://www.nsa.gov/ia/industry/crypto_elliptic_curve.cfm .

[10] 'The Arithmetic of Elliptic Curves' *Springer-Verlag, 1986.*

[11] A.V. Sutherland 'Order Computations in Generic Groups' *PhD Thesis* MIT, Submitted June 2007.

# Electronic Resources

[S ] W. Stein, SAGE - System for Algebra and Geometry Experimentation, http://sage.math.washington.edu/sage/

[W ] 'Modulo Errors' http://maths.straylight.co.uk/archives/category/phd/

# Appendix A

# SAGE class