J. Appl. Math. & Informatics Vol. **33***(2015), No. 1 - 2, pp. 145 - 155*
http://dx.doi.org/10.14317/jami.2015.145

AN EFFICIENT SEARCH SPACE IN COUNTING POINTS ON GENUS 3 HYPERELLIPTIC CURVES OVER FINITE FIELDS

GYOYONG SOHN

Abstract. In this paper, we study the bounds of the coefficients of the characteristic polynomial of the Frobenius endomorphism of the Jacobian of dimension three over a finite field. We provide explicitly computable bounds for the coefficients of the characteristic polynomial. In addition, we present the counting points algorithm for computing a group of the Jacobian of genus 3 hyperelliptic curves over a finite field with large characteristic. Based on these bounds, we found an efficient search space that was used in the counting points algorithm on genus 3 curves. The algorithm was explained and verified through simple examples.

AMS Mathematics Subject Classification : 14H45, 14G50, 94A60.
Key words and phrases : Counting Points, Hyperelliptic Curve, Cryptography.

1. Introduction

The problem of counting points on the Jacobian of algebraic curves over finite fields is very important in constructing curve-based cryptosystems. In order to obtain cryptographically suitable curves, we must determine the number of rational points on the Jacobian. If the orders of the Jacobians are sufficiently large prime numbers, then the corresponding cryptosystems are secure against various attacks.

For a long time, the counting points algorithm on elliptic and hyperelliptic curves over finite fields has been studied by numerous researchers. Schoof provided the first polynomial time algorithm for elliptic curves in all characteristics [17]. A detailed descriptions of Schoof's algorithm and the improvements by Atkin and Elkies [4] can be found in [2] and [10]. In algebraic varieties over a finite field of small characteristic, Satoh first proposed an algorithm based on p-adic methods for elliptic curves [16]. This algorithm is asymptotically faster than the Schoof's like method.

Received June 11, 2014. Revised October 10, 2014. Accepted October 16, 2014.

For higher genus curves, the equivalent problem seems to be much more difficult. In small characteristic, there exist efficient practical algorithms for computing the number of points on the Jacobian of higher genus curves based on theoretical progression. Kedlaya generalized Satoh's method to hyperelliptic curves of any genus over finite fields of odd characteristic [9]. The AGM method generalizes to ordinary hyperelliptic curves of any genus; however, only the genus 2 case is practical. In large characteristic, Pila [14] and later Adleman and Huang [1, 8] described a theoretical generalization of Schoof's approach; however, the algorithms are not practical. Currently, only the genus 2 case of this algorithm is practical [5, 6].

Throughout this paper, we concentrate on l-adic method for genus 3 curves. The l-adic method computes the number of points modulo sufficient small primes l by working in l-torsion subgroups of the Jacobian. If the characteristic is not too large, we can use the Catier-Manin operator to get additional modular information of the Jacobian. Then, one applies Weil's bounds on the coefficient of the characteristic polynomial. Hence, the final result is determined using a search algorithm such as Pollard lambda method or baby-step giant-step (BSGS) algorithm. Matsuo, Chao, and Tsujii (MCT) proposed a BSGS algorithm that speeds up the last computational part. Recently, Gaudry-Schost presented a low memory version of MCT algorithm based on birthday paradox.

When implementing the practical counting points algorithm, it is important to determine the number of candidates on the search space. It is related to the running time of the Jacobian. In the case of genus 2 curve, Ruck provided efficient bounds of the coefficients of the characteristic polynomial [15]. Recently, Haloui presented the bounds of the coefficients for genus 3 curve cases [7].

In this paper, we investigate the bounds of the coefficients of the characteristic polynomial of the Frobenius endomorphism of the Jacobian of dimension three over a finite field. In addition, we provide efficient computable bounds of the coefficients of the characteristic polynomial. These bounds will be used in the search method part of the counting points algorithm. Moreover, we derive the number of the search space for the counting points algorithm for the genus 3 curve. Based on this, we describe an algorithm that compute the order of the Jacobian group for genus 3 hyperelliptic curves over finite fields with a large characteristic. In the algorithm, we propose a reduced search space based on the experimental results. Finally, we verify the usefulness of the proposed method through some simple examples.

2. Basic Facts

Let $\mathbb{F}_q$ be a finite field of $q = p^n$ elements, where p is an odd prime. The hyperelliptic curve C of genus g over $\mathbb{F}_q$ is given by

$$C \ : \ y^2 = f(x),$$

where $f(x)$ is a polynomial in $\mathbb{F}_q[x]$ of degree $2g+1$ without singular points. We denote the Jacobian variety of a hyperelliptic curve C by J_C. Then, $J_C(\mathbb{F}_q)$ is the group of $\mathbb{F}_q$-rational points on J_C.

The *zeta function* $\zeta(t, C)$ of C can be written as

$$\zeta(t, C) = \frac{L(t, C)}{(1-t)(1-qt)},$$

where $L(t, C)$ is the *L-polynomial* of the curve. Let π_q be the Frobenius endomorphism of C and $\chi_C(t)$ the characteristic polynomial of π_q on the Tate module $T_l(J_C) \otimes \mathbb{Q}_l$. Then $\chi_{\pi_q,C}(t) = t^{2g}L(1/t, C)$. For simplicity, we write $\chi(t)$ instead of $\chi_{\pi_q,C}(t)$ if the reference to the curve is clear. The polynomial $\chi(t)$ is also called a *Weil polynomial* if the complex roots of $\chi(t)$ have the absolute value $\sqrt{q}$.

Throughout this paper, we consider the hyperelliptic curves of genus 3 over finite fields $\mathbb{F}_q$. Then, its characteristic polynomial has the form

$$\chi(t) = t^6 - s_1t^5 + s_2t^4 - s_3t^3 + qs_2t^2 - q^2s_1t + q^3, \tag{1}$$

for certain integers s_1, s_2, and s_3. We obviously have $\sharp J_C(\mathbb{F}_q) = \chi(1)$, i.e.,

$$\sharp J_C(\mathbb{F}_q) = 1 + q^3 - s_1(1+q^2) + s_2(1+q) - s_3. \tag{2}$$

Let $M_r = (q^r + 1) - N_r$, where N_r is the number of $\mathbb{F}_{q^r}$-rational points on C for $r = 1, 2, 3$. Then, we have

$$s_1 = M_1,\ s_2 = \frac{1}{2}\Big(M_1^2 - M_2\Big),\ s_3 = \frac{1}{3}\Big(M_3 - \frac{3}{2}M_2M_1 + \frac{1}{2}M_1^3\Big).$$

Thus, to compute the number of $\mathbb{F}_q$-rational points on J_C, we need only the values of three coefficients of the characteristic polynomial or, equivalently, the number of points N_r for $r = 1, 2, 3$.

The following is a well-known inequality, the Hasse-Weil bound, that bound $\sharp J_C(\mathbb{F}_q)$:

$$\lceil(\sqrt{q}-1)^{2g}\rceil \le \sharp J_C(\mathbb{F}_q) \le \lfloor(\sqrt{q}+1)^{2g}\rfloor.$$

Then, we have

$$|s_1| \le 6\sqrt{q},\ |s_2| \le 15q,\ |s_3| \le 20q\sqrt{q}. \tag{3}$$

3. The Sharp Bounds of the Coefficients of $\chi(t)$

In this section we investigate the efficient bounds of the coefficients of the characteristic polynomial $\chi(t)$. S. Haloui [7] reported on the set of characteristic polynomials of abelian varieties of dimension 3 over finite fields.

Theorem 3.1 ([7]). *Let $\chi(t) = t^6 - s_1t^5 + s_2t^4 - s_3t^3 + qs_2t^2 - q^2s_1t + q^3$ be a polynomial with integer coefficients. Then $\chi(t)$ is a Weil polynomial if and only if the following conditions hold*

- $|s_1| \le 6\sqrt{q}$,
- $4\sqrt{q}|s_1| - 9q \le s_2 \le \frac{s_1^2}{3} + 3q$,

- $-\frac{2s_1^3}{27}+\frac{s_1s_2}{3}+qs_1-\frac{2}{27}(s_1^2-3s_2+9q)^{3/2}\le s_3\le -\frac{2s_1^3}{27}+\frac{s_1s_2}{3}+qs_1+\frac{2}{27}(s_1^2-3s_2+9q)^{3/2}$,
- $-2qs_1-2\sqrt{q}s_2-2q\sqrt{q}\le s_3\le -2qs_1+2\sqrt{q}s_2+2q\sqrt{q}$.

Proof. See S Haloui [7]. □

Denote U_{3a} (resp. L_{3a}) the upper (resp. lower) bound of s_3 in (3) of Theorem 3.1, respectively, and U_{3b} (resp. L_{3b}) the upper (resp. lower) bound of s_3 in (4) of Theorem 3.1, respectively. The following theorem gives an efficient choice between the two upper (resp. lower) bounds for s_3 in Theorem 3.1.

Theorem 3.2. *Let $\chi(t)$ be a Weil polynomial of degree 6 defined by equation (1). Then, the upper bound of s_3 is defined as*

$$s_3\le -2qs_1+2\sqrt{q}s_2+2q\sqrt{q} \text{ if } s_2\le t(s_1),$$
$$s_3\le -\frac{2s_1^3}{27}+\frac{s_1s_2}{3}+qs_1+\frac{2}{27}(s_1^2-3s_2+9q)^{3/2} \text{ otherwise}$$

where $t(s_1)=s_1^2/4+\sqrt{q}s_1$. Similarly, the lower bound of s_3 is defined as

$$-2qs_1-2\sqrt{q}s_2-2q\sqrt{q}\le s_3 \text{ if } s_2\le r(s_1),$$
$$-\frac{2s_1^3}{27}+\frac{s_1s_2}{3}+qs_1-\frac{2}{27}(s_1^2-3s_2+9q)^{3/2}\le s_3 \text{ otherwise.}$$

where $r(s_1)=s_1^2/4-\sqrt{q}s_1$.

Proof. First, we consider the upper bounds U_{3a} and U_{3b} of s_3. The difference between U_{3a} and U_{3b} is as follows:

$$U_{3a}-U_{3b}=-\frac{1}{27}s_1^3+\frac{s_1s_2}{3}+3qs_1-2\sqrt{q}s_2-2q\sqrt{q}+\frac{2}{27}(s_1^2-3s_2+9q)^{3/2}.$$

If $U_{3a}-U_{3b}=0$, then we have a line of intersection for them. After squaring the equation, we have the following

$$\frac{1}{27}(4\sqrt{q}s_1+s_1^2-4s_2)(9q-4\sqrt{q}s_1+s_2)^2=0.$$

It is imply that $4\sqrt{q}s_1+s_1^2-4s_2=0$ or $9q-4\sqrt{q}s_1+s_2=0$. Since the left hand side of the second equation $9q-4\sqrt{q}s_1+s_2=0$ is the lower bound of s_2 in (2) of Theorem 3.1, the line of intersection of U_{3a} and U_{3b} in the possible range (s_1,s_2) is $s_2=s_1^2/4+\sqrt{q}s_1$. Thus we can easily check if $s_2<s_1^2/4+\sqrt{q}s_1$, then $U_{3a}-U_{3b}$ is positive, and if $s_2>s_1^2/4+\sqrt{q}s_1$, then $U_{3a}-U_{3b}$ is negative.

Similarly, we compute the difference between L_{3a} and L_{3b}, and if its difference is zero, then we have the following:

$$\frac{1}{27}(4\sqrt{q}s_1-s_1^2+4s_2)(9q+4\sqrt{q}s_1+s_2)^2=0.$$

Then the equation $s_2=s_1^2/4-\sqrt{q}s_1$ is contained within the range of s_1 and s_2. It is the line of intersection for L_{3a} and L_{3b}. Thus, we have if $L_{3a}-L_{3b}>0$, then $s_2>s_1^2/4-\sqrt{q}s_1$, or otherwise, $s_2<s_1^2/4-\sqrt{q}s_1$. The proof is completed. □

Lemma 3.3. *If $s_1 \in [2\sqrt{q}, 6\sqrt{q}]$, then the lower bound of s_3 in (1) is defined as L_{3a}. If $s_1 \in [-6\sqrt{q}, -2\sqrt{q}]$, then the upper bound of s_3 is defined as U_{3a}.*

Proof. For $s_1 \in [2\sqrt{q}, 6\sqrt{q}]$, we have $r(s_1) < 4\sqrt{q}s_1 - 9q$. Because of the previous theorem, the lower bound of s_3 is just defined as L_{3a}. Similarly, we can check the upper bound of s_3 for $s_1 \in [-6\sqrt{q}, -2\sqrt{q}]$. □

Now we discuss the inequalities of s_3 in $|s_1| \leq 2\sqrt{q}$. The following theorem shows the sharp bound of s_2 with respect to s_1.

Theorem 3.4. *Let $\chi(t)$ be a Weil polynomial of degree 6 defined by equation (1). Then the lower bound of s_2 is defined as*

$$-q \leq s_2 \text{ if } |s_1| < 2\sqrt{q},$$
$$4\sqrt{q}|s_1| - 9q \leq s_2 \text{ if } |s_1| \geq 2\sqrt{q}.$$

Proof. If $\chi(t)$ is a Weil polynomial of degree 6 defined by equation (1), then each coefficient of the polynomial $\chi(t)$ should satisfy the inequalities of s_i in Theorem 3.1 for $i = 1, 2, 3$.

Note that $U_{3a} = L_{3a} = \frac{1}{27}s_1^3 + 2qs_1$ at the upper boundary of $s_2 = \frac{s_1^2}{3} + 3q$ for all $s_1 \in [-6\sqrt{q}, 6\sqrt{q}]$. For $s_1 \geq 0$, the value of U_{3b} is equal to L_{3a} at the lower boundary of $s_2 = 4\sqrt{q}s_1 - 9q$ for all s_1. i.e.,$U_{3b} = L_{3a}$. Similarly, $U_{3a} = L_{3b}$ at the lower boundary of $s_2 = -4\sqrt{q}s_1 - 9q$ for $s_1 \leq 0$. From Theorem 3.2 and Lemma 3.3, in $|s_1| \geq 2\sqrt{q}$, the upper bounds of s_3, U_{3a} and U_{3b}, are always bigger than the lower bounds of s_3, L_{3a} and L_{3b}.

Let us consider $|s_1| \leq 2\sqrt{q}$. If $s_2 \geq r(s_1)$ for $s_1 \in [0, 2\sqrt{q}]$ and $s_2 \geq t(s_1)$ for $s_1 \in [-2\sqrt{q}, 0]$, then the upper bounds of s_3 are always bigger than the lower bounds of s_3. Now, let us consider $4\sqrt{q}s_1 - 9q \leq s_2 \leq r(s_1)$ and $-4\sqrt{q}s_1 - 9q \leq s_2 \leq t(s_1)$. The line of intersection for L_{3b} and U_{3b} is

$$s_3 = -2qs_1, \ s_2 = -q.$$

Thus if $s_2 < -q$, then the coefficient of s_3 is not satisfied with the inequalities of both condition (3) and (4) in Theorem 3.1. The inequality of s_3 in (4) can be changed to $U_{3b} \leq s_3 \leq L_{3b}$. Thus, the above conditions hold. □

4. Counting Points Algorithm

In this section, we describe the counting points algorithm on hyperellipitic curves of genus 3 over finite fields with a large characteristic.

4.1. MCT Algorithm. Matsuo, Chao and Tsujii present an algorithm to determine the order of the Jacobian for hyperelliptic curves over finite fields using an improved BSGS method. Now, we describe the MCT algorithm for the computational part.

Assume that the characteristic polynomial is known modulo some positive integer m, i.e., s_1, s_2 and s_3 are known modulo m. Denote that for $i = 1, 2, 3$

$$s_i = \overline{s_i} + mt_i, \tag{4}$$

with $\overline{s_i}, t_i \in \mathbb{Z}$ $(0 \leq \overline{s_i} < m)$. Note that s_i is known and t_i is unknown. We substitute (4) into (2) and denote $K = 1+q^3-\overline{s_1}(1+q^2)+\overline{s_2}(1+q)-\overline{s_3}$. Then, the order of the Jacobian follows the equation

$$\sharp J_C(\mathbb{F}_q) = K - t_1 m(1+q^2) + t_2 m(1+q) - t_3 m.$$

We should determine the values (t_1, t_2, t_3) in order to get $\sharp J_C(\mathbb{F}_q)$. Hence, $\sharp J_C(\mathbb{F}_q)$ can be computed by finding the triples (t_1, t_2, t_3) such that

$$K \cdot D + (-t_1 m(1+q^2) + t_2 m(1+q) - t_3 m) \cdot D = 0, \tag{5}$$

for a random element $D \in J_C(\mathbb{F}_q)$. For some positive integer N to be specified, let u, v be integers such that

$$t_3 = u + vN,\ 0 \leq u < N. \tag{6}$$

By substituting (6) into (5), $\sharp J_C(\mathbb{F}_q)$ can be computed by finding the 4-tuples (t_1, t_2, u, v) such that

$$(K - t_1 m(1+q^2) + t_2 m(1+q) - um) \cdot D = (vNm) \cdot D, \tag{7}$$

for a random element $D \in J_C(\mathbb{F}_q)$ in the corresponding ranges. These computations are terminated by searching for a collision between the LHS and the RHS of (7) among different candidates. Moreover, the BSGS method is used in (7). The algorithm requires the computation of $O(N)$ group operations and storage of $O(N)$ group elements.

4.2. The Number of The Search Space. The value N of the previous method is determined by the number of different candidates in the searching space. From the Hasse-Weil bound, the number of the search space is $14,400q^3$. Thus, the value of N is approximately $120q^{3/2}/m^{3/2}$, which is the running time of group operations in J_C.

Then, we compute the efficient value of N from the results in Section 3. Assume that s_1 is a positive integer smaller than $6\sqrt{q}$ (for the negative part of s_1, the computation is same because the boundaries of s_i are symmetrical about s_2 in dimension 3.) Let $t(s_1) = s_1^2/4 + \sqrt{q}s_1$ and $r(s_1) = s_1^2/4 - \sqrt{q}s_1$. We compute the two parts centered by $2\sqrt{q}$ of s_1. For $s_1 \in [0, 2\sqrt{q}]$, the number of (s_1, s_2, s_3) is:

$$\int_0^{2\sqrt{q}} \Big(\int_{-q}^{r(s_1)} (U_{3b} - L_{3b}) ds_2 + \int_{r(s_1)}^{t(s_1)} (U_{3b} - L_{3a}) ds_2 + \int_{t(s_1)}^{\frac{s_1^2}{3}+3q} (U_{3a} - L_{3a}) ds_2 \Big) ds_1 = \frac{448}{45} q^3.$$

For $s_1 \in [2\sqrt{q}, 6\sqrt{q}]$, the number of (s_1, s_2, s_3) is:

$$\int_{2\sqrt{q}}^{6\sqrt{q}} \Big(\int_{4\sqrt{q}s_1 - 9q}^{t(s_1)} (U_{3b} - L_{3a}) ds_2 + \int_{t(s_1)}^{\frac{s_1^2}{3}+3q} (U_{3a} - L_{3a}) ds_2 \Big) ds_1 = \frac{64}{45} q^3.$$

Thus the number of the triples (s_1, s_2, s_3) is roughly $\frac{1024}{45}q^3$. Moreover, the number S of the (t_1, t_2, t_3) is

$$S = \frac{1024 q^3}{45 m^3}.$$

We can choose a value of N as

$$N \approx \sqrt{S} = \frac{32q^{3/2}}{5m^{3/2}} = 6.4q^{3/2}/m^{3/2}.$$

Thus the algorithm requires the computation of $O(N)$ point multiples and memory storage. Hence, we see that the number of the search space is reduced to a constant factor of about 18 compared with the Weil's bound.

4.3. Gaudry-Schost Algorithm. Gaudry-Schost algorithm is a low-memory algorithm for computing the number of the Jacobian of hyperelliptic curves over finite fields [6]. The basic idea is the same as the kangaroo algorithm of Pollard in the van Oorschot and Wiener formulation [13]. We now briefly describe the Gaudry-Schost algorithm for genus 3 curves.

Let BU_i (rep. BL_i) be an upper (rep. lower) bound of t_i for $i = 1, 2, 3$. From (5), we wish to find integers $(t_1, t_2, t_3) \in \mathbb{Z}^3$, $t_i \in [BL_i, BU_i]$, such that

$$\overline{D} = t_1 \cdot D_1 + t_2 \cdot D_2 + t_3 \cdot D_3, \tag{8}$$

where $\overline{D} = K \cdot D$, $D_1 = (1 + q^2) \cdot m \cdot D$, $D_2 = -(1 + q) \cdot m \cdot D$, and $D_3 = m \cdot D$ for a random element $D \in J_C(\mathbb{F}_q)$. Denote the set

$$V = \{(n_1, n_2, n_3) \in \mathbb{Z}^3 \mid n_i \in [BL_i, BU_i] \text{ for } i = 1, 2, 3\}$$

and $M_i = \lfloor (BL_i + BU_i)/2 \rfloor$ for $i = 1, 2, 3$. The basic Gaudry-Schost algorithm for this problem is as follows. The tame set is defined as

$$T = \{(n_1, n_2, n_3) \in \mathbb{Z}^3 \mid (n_1, n_2, n_3) \in V\},$$

and the wild set as

$$\begin{aligned} W &= (t_1 - M_1, t_2 - M_2, t_3 - M_3) + T \\ &= \Big\{(t_1 - M_1 + n_1, t_2 - M_2 + n_2, t_3 - M_3 + n_3) \in \mathbb{Z}^3 \mid (n_1, n_2, n_3) \in T\Big\}. \end{aligned}$$

The Gaudry and Schost algorithm run a large number of (deterministic) pseudorandom walk. Half the walks are "tame walks", which means that every element in the walk is of the form $a_1 \cdot D_1 + a_2 \cdot D_2 + a_3 \cdot D_3$ where the integer triples $(a_1, a_2, a_3) \in T$ (though note that with very small probability some walks will go outside T). The other half are "wild walks", which means that every element is of the form

$$\widetilde{D} + b_1 \cdot D_1 + b_2 \cdot D_2 + b_3 \cdot D_3,$$

where the integer triples $(b_1, b_2, b_3) \in T$ and

$$\widetilde{D} = \overline{D} - M_1 \cdot D_1 - M_2 \cdot D_2 - M_3 \cdot D_3.$$

Each walk proceeds until a distinguished point is hit. This distinguished point is then stored on a server, together with the corresponding exponent vectors (n_1, n_2, n_3) and a flag indicating which sort of walk it was. The collusion of two different types of walks exist if and only if $a_i - b_i = t_i - \lfloor \frac{BL_i - BU_i}{2} \rfloor$ for all i. Then, we can find the triples (t_1, t_2, t_3) such that $\chi(1) \cdot D = 0$. Since a collision

between the tame and wild walks can only occur in $T \cap W$, we can easily check that $\sharp(T \cap W) \in [\frac{\sharp V}{8}, \sharp V]$. The following theorem shows the running time of the Gaudry-Schost algorithm in the case of genus 3 curve.

Theorem 4.1. *Given the problem by* (8) *and N is the number of the search space. The expected number of group operations in the average case of the Gaudry-Schost algorithm is* $2.85\sqrt{N}$.

Proof. We assume that $BL_i = -BU_i$ and denote B_i for each i. Then, we consider the following problem instance: $\overline{D} = t_1 \cdot D_1 + t_2 \cdot D_2 + t_3 \cdot D_3$ with $t_i \in [-B_i, B_i]$ for i. The number of overlaps between T and W is

$$\sharp(T \cap W) = \prod_{i=1}^{3}(2B_i + 1 - |t_i|).$$

Therefore, we expect only about $\sharp(T \cap W)/N$ of the walks to be in $T \cap W$. The algorithm is based on the birthday paradox. Assume that t_1, t_2 and t_3 are uniformly distributed. Thus the average case expected running time is

$$\frac{2^3}{N}\int_{t_1=0}^{B_1}\int_{t_2=0}^{B_2}\int_{t_3=0}^{B_3}\Big(\frac{\sharp(T \cap W)}{N}\Big)^{-\frac{1}{2}}\sqrt{\pi N}dt_1dt_2dt_3 = 2^6\sqrt{\pi}\prod_{i=1}^{3}(\sqrt{2B_i+1} - \sqrt{B_i+1})$$
$$\approx 2^6\Big(1 - \frac{1}{\sqrt{2}}\Big)^3\sqrt{\pi N} = (4 - 2\sqrt{2})^3\sqrt{\pi N} \approx 2.85\sqrt{N}.$$

□

If χ is the known modulo m with $m < 12\sqrt{q}$, then there are many candidates for s_1, s_2 and s_3 which have bounds in Theorem 3.1. These bounds yield the approximate bounds for t_1, t_2 and t_3:

$$BU_1 - BL_1 = 12\sqrt{q}/m,\ BU_2 - BL_2 = 16q/m,\ BU_3 - BL_3 = 40q\sqrt{q}/m. \quad (9)$$

Hence, the number of the search space is $\sharp V = 7680q^3/m^3$. From Theorem 4.1, the approximate value of 2.85 then yield a running time of about $249.761q^{3/2}/m^{3/2}$ operations in $J_C(\mathbb{F}_q)$. Compared with the MCT algorithm we lost a constant factor of about 39.

In the case where $m \geq 12\sqrt{q}$, the coefficient s_1 is uniquely determined since $|s_1| \leq 6\sqrt{q} \leq m/2$. Then we only have to consider the corresponding bounds for t_2 and t_3 in (9). Hence, the number of the search space is $\sharp V = 640q^{5/2}/m^2$. In [6], the average case running time for a genus 2 curve is approximately $2.43\sqrt{N}$. Thus the expected running time is $61.47q^{5/4}/m$ group operations in $J_C(\mathbb{F}_q)$. This is worse than the $O(q^{3/2}/m^{3/2})$ complexity.

4.4. Reducing the Search Space. In this section, we investigate the distribution of the coefficients (s_1, s_2, s_3) in the search space. It was up to us to reduce the space complexity for the counting points algorithm. We first selected 10,000 random monic, square-free polynomials of degree 7 over finite fields $\mathbb{F}_p$ with the small random prime p. Then, we computed the values (s_1, s_2, s_3) for

the corresponding curves using a well-known algorithm. Table 4.4 shows the proportion of the curves for which each s_i is larger than some bound.

TABLE 1. Distribution of (s_1, s_2, s_3)

	s_1		s_2			s_3		
Bounds of $\lvert s_i\rvert$	$\geq 3\sqrt{q}$	$\geq 2\sqrt{q}$	$\geq 5q$	$\geq 4q$	$\geq 3q$	$\geq 8q\sqrt{q}$	$\geq 6q\sqrt{q}$	$\geq 4q\sqrt{q}$
Proportion	0.242	4.741	0.298	1.143	4.058	0.0186	0.255	1.734

In Table 1, for $|s_3| \geq 8q\sqrt{q}$, the proportion of the curves is approximately 0.0186. So, more than 99.9% of the curves have $|t_3| < 8q\sqrt{q}/m$. Thus, we can restrict the search space to the following bounds:

$$V = \{(n_1, n_2, n_3) \mid n_1 \in V_1, n_2 \in V_2, n_3 \in V_3\},$$

where $V_1 = [-4\sqrt{q}/m, 4\sqrt{q}/m], V_2 = [-q/m, 7q/m]$, and $V_3 = [-8q\sqrt{q}/m, 8q\sqrt{q}/m]$.

In this case, for the fixed value t_3, we can easily obtain the bounds of V_1 and V_2 according to the bounds in Section 3. Moreover, the overlapping factor of W and T is at least $1/4$. The number of elements in V is reduced to $1024q^3/m^3$ so that the expected runtime is approximately $91.2q^{3/2}/m^{3/2}$ group operations. Therefore, we reduced the running time by a factor of 2.7 compared to the previous running time.

For $|t_3| < 6q\sqrt{q}$, 99.4% of the curves exists in this bound. In this case, the overlapping factor is at least $\frac{11}{192} = 0.057$. If $|t_3| > 16q\sqrt{q}/m$ or, $|t_3| > 14q\sqrt{q}/m$ and $|s_2| > 11q$, the sets, W and T, do not overlap.

5. Experimental Results

We implemented two algorithms in C++ using Shoup's NTL library on a Pentium 2.13 GHz computer with less than 2 GB memory.

Example 5.1. Let prime $p = 10^6 + 37$ and C be the hyperelliptic curve given by

$$f(x) = x^7 + 168985x^6 + 145758x^5 + 68532x^4 + 69904x^3 + 54646x^2 + 17958x + 33627.$$

Using the classical counting points algorithm we easily computed the values of s_1, s_2 and s_3 modulo $m = 1874$. The curve is such that $s_1/\sqrt{p} \approx -0.35$, $s_2/p \approx 0.27$ and $s_3/p\sqrt{p} \approx -0.55$, therefore it is not close to any border of the bounds. The group of points has cardinality

$$\sharp J_C(\mathbb{F}_p) = 1000465307750115976.$$

Example 5.2. Let $p = 26144785074025909$ be a 55-bit prime and let C be the curve defined by $C : y^2 = x^7 + 4857394849x$. The group order of the Jacobian is given by:

$$178712622571907053989539231112397193490170498152 84$$

The number of the Jacobian is of 163 bits and the total time is 259 s.

6. Conclusions

In this paper, we have presented algorithms for computing the orders of the Jacobian varieties of genus 3 hyperelliptic curves over finite fields. We also have computed the efficient bound of the characteristic polynomial of the Jacobian and determined the number of the search space. From these bounds, we have given the number of the search space, equivalently, the number of candidates in the counting points algorithm. We also studied the search space with the practical algorithm. Finally, we presented simple examples of random hyperelliptic curves of genus 3 over finite fields.

References

1. L. Adleman and M. D. Huang, Counting rational points on curves and abelian varieties over finite fields, in H. Cohen (ed.), ANTS-II, LNCS 1122, Springer-Verlag, (1996) pp. 1–16.
2. I. Blake, G. Seroussi and N. Smart, Elliptic curves in cryptography, London Math. Soc. Lecture Note Series 265 (1999).
3. J. Denef and F. Vercauteren, An Extension of Kedlaya's Algorithm to Hyperelliptic Curves in Characteristic 2, J. Cryptology, **19** (2006), 1–25.
4. N. Elkies, Elliptic and modular curves over finite fields and related computational issues, Computational perspectives on number theory, vol. 7 of AMS/IP Stud. Adv. Math., pp.21–76, Am. Math. Soc. (1998).
5. P. Gaudry and R. Harley, Counting points on hyperelliptic curves over finite fields, ANTS-IV, W. Bosma ed., LNCS 1838 (2000), Springer-Verlag, 297–312.
6. P. Gaudry and E. Schost, A low-memory parallel version of Matsuo, Chao and Tsujii's algorithm, In D. A. Buell, editor, Proceedings of Algorithm Number Theory Sympositum-ANTS VI. volume 3076 of LNCS, pages 208-222, Springer-Verlag, 2004.
7. S. Haloui, The characteristic polynomials of abelian varieties of dimensions 3 over finite fields, J. Number theory, 2011.
8. M. D. Huang and D. Ierardi, Counting points on curves over finite fields, J. Symb. Comp. **25**(1) (1998), 1-21.
9. K.S. Kedlaya, Counting points on hyperelliptic curves using Monsky-Washnitzer cohomology, J. Ramanujan Math. Soc. **16** (2001), 323-338.
10. R. Lercier, Algorithmique des courbes elliptiques dans les corps finis. Thése, École polytechnique, June 1997.
11. Yu. I. Manin, The Hasse-Witt matrix of an algebraic curve, AMS Trans. Ser. **2**(45) (1965), 245–264.
12. K. Matsuo, J. Chao, and S. Tsujii, An improved baby step giant step algorithm for point counting of hyperelliptic curves over finite fields, In C. Fiecker and D. Kohel, editors, Proceedings of Algorithm Number Theory Symposium-ANTS V. volume 2369 of LNCS, pages 461–474, Springer-Verlag, 2002.
13. V. Oorschot and P.C., Wiener, M.J., Parallel collusion Search with Cryptanalytic Applications, J. Cryptology **12** (1990), 1–28.
14. J. Pila, Frobenius maps of abelian varieties and finding roots of unity in finite fields, Math. Comp. **55**(192) (1990), 745–763.
15. H.-G.Rück, Abelian surfaces and jacobian varieties over finite fields, Compositio Math. **76**(3) (1990), 351–366.
16. T. Satoh, The canonical lift of an ordinary elliptic curve over a finite field and its point counting, J. Ramanujan Math. Soc. **15** (2000), 247–270.

17. R. Schoof, Elliptic curves over finite fields and the computation of square roots mod p, Math. Comp. **44** (1985) 483–494.
18. G. Sohn, Pointing algorithm for one-dimensional family of genus 3 nonhyperelliptic curves over finite fields, J. Appl. Math. & Informatics **30** (2012), 101–109.

Gyoyong Sohn received M.Sc. and Ph.D from Kyungpook National University. He is currently an assistant professor in the Department of Mathematics Education at Deagu National University of Education. His research interests are computational algebraic geometry and cryptography.

Department of Mathematics Education, Deagu National University of Education, 219 Jungang-daero, Deagu 705-715, Korea.
e-mail: gysohn@dnue.ac.kr