

PAPER

An Addition Algorithm in Jacobian of C_{34} Curve

Seigo ARITA[†], *Member*

SUMMARY This paper gives an efficient algorithm to compute addition in Jacobian of C_{34} curves, aiming at C_{34} curve cryptosystems. Using C_{34} curves for cryptosystems has two advantages. The first is safety and the second is the short size of the base field. In the paper, we modify the addition algorithm of [1] for C_{ab} curves in the specific manner to C_{34} curves. We classify all of the forms of the Groebner bases of ideals involved in the algorithm and eliminate the use of Buchberger algorithm from it. Our resulting algorithm computes the addition in Jacobian of C_{34} curves in about 3 times amount of computation of the one in elliptic curves, when the sizes of groups are set to be the same.

key words: algebraic curve cryptosystem, C_{34} curve, C_{ab} curve, addition in Jacobian, Groebner base

1. Introduction

Although now elliptic curve cryptosystems are widely used, algebraic curve cryptosystems with Jacobian group of more general non-elliptic curves, such as hyperelliptic, superelliptic[5] and C_{ab} curve[1], has not been used. One of the main reasons for it has been the heavy computational load of addition in Jacobian of such non-elliptic curves.

Surprisingly, Harley[8], by carefully optimizing Cantor's algorithm[3], gives an algorithm for addition in Jacobian of hyperelliptic curves of genus two, which computes the addition on the hyperelliptic curve almost in the same time as the addition on elliptic curves, when the sizes of groups are set to be the same. Harley's algorithm is modified by Matsuo and Chao[13] and by Lange[11], and is generalized to the case of genus three hyperelliptic curves by Kuroki et al.[10], Pelzl et al.[15] and Gonda et al.[6]. So, now it is said that hyperelliptic curve cryptosystems has practical efficiency.

In this paper, which is an extended version of [2], we treat C_{34} curves. C_{34} curves are genus three non-hyperelliptic curves, being special cases of C_{ab} curves[12],[14]. Using C_{34} curves for cryptosystems has two advantages. The first is safety. Since C_{34} curves have different mathematical features from elliptic or hyperelliptic curves, even if we encountered some new attack method which seriously damages of safety of elliptic or hyperelliptic curve cryptosystems, C_{34} curves possibly would survive. The second advantage is the short size of the base field. Since C_{34} curve has genus three, the size of base field for C_{34} curve cryptosystems

is only a third of the one for elliptic curve cryptosystems. If we use 64-bit CPU, C_{34} curve cryptosystems don't require multi-precision package.

In order to realize a practical C_{34} curve cryptosystem, we need to construct an addition algorithm for Jacobian of C_{34} curves which is as efficient as Harley's algorithm. In the paper, we modify the addition algorithm of [1] for C_{ab} curves in the specific manner to C_{34} curves. We classify all of the forms of the Groebner bases of ideals involved in the addition in Jacobian of C_{34} curve. This classification eliminates the use of Buchberger algorithm from the algorithm of [1]. We show that our resulting algorithm computes the addition in Jacobian of C_{34} curves in about 3 times amount of computation of the one in elliptic curves, when the sizes of groups are set to be the same. This gives the first step toward an addition algorithm for Jacobian of C_{34} curves which is as efficient as Harley's algorithm.

2. C_{34} Curve and Its Jacobian Group

C_{34} curve, which is a special case of C_{ab} curve found by Miura[12],[14], is a nonsingular plan curve defined by the following form of polynomial F :

$$F = Y^3 + a_0X^4 + a_1XY^2 + a_2X^2Y + a_3X^3 + a_4Y^2 + a_5XY + a_6X^2 + a_7Y + a_8X + a_9, \quad (1)$$

where a_i 's are elements of the definition field k and $a_0 \neq 0$.

C_{34} curve C has a unique point ∞ at the infinity. The function Y and X has the unique pole at ∞ of order four and three, respectively. We can see the gap sequence at ∞ is $\mathbb{N}_0 - \langle 3, 4 \rangle = \{1, 2, 5\}$, and the genus of C_{34} is found to be three.

Let $D_C^0(k)$ denote the group of divisors of degree 0 on C defined over k , and $P_C(k)$ be the group of principal divisors on C defined over k . As well known, Jacobian group $J_C(k)$ on C is defined to be the factor:

$$J_C(k) = D_C^0(k)/P_C(k).$$

On the other hand, let $R = k[X, Y]/F$ be the coordinate ring of C . Since C_{34} curve C is nonsingular by the definition, R is integrally closed domain, so R is a Dedekind domain. Hence, all of the nonzero fractional

[†]The author is with Institute of Information Security, Kanagawa, Japan.

ideals of R compose a group $I_R(k)$. Putting the group of principal ideals of R $P_R(k)$, the ideal class group $H_R(k)$ of R is defined to be the factor:

$$H_R(k) = I_R(k)/P_R(k).$$

In general, for a nonsingular curve, we can identify divisors on the curve and ideals of the coordinate ring, and its Jacobian group $J_C(k)$ is naturally isomorphic to the ideal class group $H_R(k)$ (Example 6.3.2 of [9]):

$$J_C(k) \cong H_R(k) \\ [E - n\infty] \mapsto \left[\bigcup_{n=0}^{\infty} L(m\infty - E) \right],$$

where E is a positive divisor prime to ∞ .

Ideals are more useful than divisors to implement algorithms. From now on, we treat Jacobian group $J_C(k)$ as the ideal class group $H_R(k)$ of the coordinate ring R .

3. Preparations for Groebner Bases

Here, we make preparations for Groebner bases of ideals. For details, see [4].

Let ' $<$ ' be an well-order among monomials in a polynomial ring $S = k[X_1, \dots, X_n]$. When the order ' $<$ ' is compatible with the product in the sense that we have $M_1M_3 < M_2M_3$ whenever $M_1 < M_2$, the order ' $<$ ' is called a monomial order. In the rest of this section, we suppose that any monomial order ' $<$ ' is given and fixed for a polynomial ring S .

For a polynomial f in S , the largest monomial, with respect to the monomial order ' $<$ ', appearing in f is called a leading monomial of f and denoted by $\text{LM}(f)$. For an ideal I of S , the ideal generated by all of the leading monomials of polynomials in I is denoted by $\text{LM}(I)$. Suppose an ideal $I = (f_1, \dots, f_s)$ of S generated by f_1, \dots, f_s is given. The set $\{f_1, \dots, f_s\}$ is called Groebner base of I when it satisfies

$$\text{LM}(I) = (\text{LM}(f_1), \dots, \text{LM}(f_s)).$$

Let I be an ideal of S . The set of monomials (or their multi-degrees) not belonging to $\text{LM}(I)$ is called a Δ -set of I and denoted by $\Delta(I)$. $\Delta(I)$ gives a basis of the vector space S/I over k . When we plot monomials $X_1^{m_1} X_2^{m_2} \dots$, or their multi-degrees (m_1, m_2, \dots) in $\Delta(I)$ on the (m_1, m_2, \dots) -space, there appears a convex set, of which surrounding lattice points correspond to leading monomials of polynomials in Groebner base of I .

Let $R = S/F$ be a coordinate ring of a C_{34} curve defined by F . By identifying ideals of R with ideals of S including F , we can consider Groebner bases for ideals of R . For a 0-dimensional ideal I (i.e. the zero set of I is finite), we define its order $\delta(I)$ as

$$\delta(I) = \dim_k R/I.$$

By the definition, we see $\delta(I) = \sharp\Delta(I)$. Since C_{34} curve is nonsingular, $\delta(IJ) = \delta(I)\delta(J)$. If $I = (f)$ is a principal ideal in R , we have $\delta(I) = -v_\infty(f)$.

4. An Addition Algorithm in Jacobian of C_{34} curve — abstract level

Let $R = k[X, Y]/F$ be a coordinate ring of a C_{34} curve C defined by the polynomial F (1). We can define a monomial order ' $>$ ', called C_{34} order, by the pole number of monomials at ∞ . That is,

$$X^{m_1} Y^{n_1} > X^{m_2} Y^{n_2} \stackrel{\text{def}}{\iff} 3m_1 + 4n_1 > 3m_2 + 4n_2 \\ \text{or } 3m_1 + 4n_1 = 3m_2 + 4n_2, m_1 < m_2$$

Hereafter, we always use C_{34} order to compare monomials in R .

For an ideal I in R , let f_I be the nonzero 'monic' polynomial with the smallest leading monomial in I . We define I^* as

$$I^* = (f_I) : I \quad (= \{g \in R \mid gI \subset (f_I)\}).$$

Then, we have

Proposition 1 *Let I, J be any ideals in the coordinate ring R . We have*

- (1) I is equivalent to I^{**} .
- (2) I^{**} is an ideal equivalent to I with the smallest order.
- (3) If I and J are equivalent, then we have $I^* = J^*$. In particular, $I^{**} = (I^{**})^{**}$.

Proof (1) I^* is equivalent to the inverse ideal of I from definition.

(2) Let J be an (integral) ideal equivalent to I^{-1} . There is a $f \in R$ satisfying $J I = (f)$. From the definition of I^* , $I^* I = (f_I)$. So, we have

$$\delta(J)\delta(I) - \delta(I^*)\delta(I) = -v_\infty(f) + v_\infty(f_I) \geq 0,$$

by the definition of f_I . Therefore, I^* is an (integral) ideal equivalent to I^{-1} with the smallest order. So, I^{**} is an (integral) ideal equivalent to I with the smallest order.

(3) If I and J are equivalent, there are $j, h \in R$ satisfying $J = \frac{j}{h}I$. Then, we have $f_J = \frac{j}{h}f_I$. So, for $g \in R$,

$$gJ \subset (f_J) \iff g \frac{j}{h}I \subset \left(\frac{j}{h}f_I\right) \iff gI \subset (f_I)$$

□

An ideal I in the coordinate ring R is called *reduced* when we have $I^{**} = I$. By Proposition 1(1),(3), any ideal in R is equivalent to the unique reduced ideal. That is, reduced ideals compose a complete representative system of ideal classes. Moreover, by Proposition 1(2), we see that a reduced ideal has the smallest order among ideals in the same ideal class. This property should be a merit to implement algorithms.

Using reduced ideals as a representative system of ideal classes, we get the following addition algorithm in Jacobian of C_{34} curve.

Algorithm 1 (Addition in Jacobian of C_{34} curve)

Inputs: reduced ideals I_1, I_2 in the coordinate ring R

Output: reduced ideal I_3 equivalent to the ideal product $I_1 \cdot I_2$

1° $J \leftarrow I_1 \cdot I_2$

2° $J^* \leftarrow (f_J) : J$

3° $I_3 \leftarrow (f_{J^*}) : J^*$

We note that Harasawa and Suzuki[7] also gives an addition algorithm on Jacobian of C_{ab} curves, by extending the addition algorithm on superelliptic curves of Galbraith, Paulus, and Smart[5]. Their algorithms use LLL-algorithm to reduce ideals. Although [7] gives an asymptotic evaluation of the amount of computation of their algorithm, the evaluation of the O -constants is not given.

5. Ideal Classification

In this section, we classify ideals appearing in performing Algorithm 1, in order to implement Algorithm 1 efficiently. Since the genus of C_{34} curve is three, the orders of those ideals are not greater than six. So, it is sufficient to classify ideals of R with order not greater than six.

Hereafter, even if the defining polynomial F (Equation (1)) of C_{34} curve C appears in Groebner base of an ideal, we do not explicitly show it, and a_i, b_j, c_k denote coefficients of polynomials in Groebner bases.

5.1 Ideals of order 6

Let I be an ideal in R of order 6. By the definition of order, $V = R/I$ is a sixth dimensional vector space over the definition field k .

Type 61

An ideal I of order six has six zero points including multiplicities. When those six points are in ‘general’ positions, the first six monomials $1, X, Y, X^2, XY, Y^2$ with respect to the C_{34} order are linearly independent on those six points. So, the set of monomials $M =$

$\{1, X, Y, X^2, XY, Y^2\}$ is a basis of the vector space $V = R/I$. In this case, we call I an ideal of *type 61*.

It is easily seen that the fact that the set of monomials M is linearly dependent in $V = R/I$ is equivalent to the fact that there is a monomial in M belonging to $\text{LM}(I)$. So, If I is an ideal of type 61, then the set of monomials M is nothing but $\Delta(I)$. Using notation of multi-degrees, we have $\Delta(I) = \{(0,0), (1,0), (0,1), (2,0), (1,1), (0,2)\}$. It is easily seen that lattice points surrounding $\Delta(I)$ are $\{(0,3), (1,2), (2,1), (3,0)\}$. So, Groebner base of an ideal I of type 61 has the form in Table 1. Those three polynomials correspond to the lattice points $(3,0), (2,1), (1,2)$ (Note the lattice point $(0,3)$ corresponds to the defining polynomial F).

Type 62 and 63

In general, six monomials $1, X, Y, X^2, XY, Y^2$ are not linearly independent in $V = R/I$. First, we consider the case that the first five monomials $1, X, Y, X^2, XY$ with respect to the C_{34} order are linearly independent, but the sixth monomial Y^2 is equal to a linear sum of them in V .

In that case, $\Delta(I)$ is a convex set of order 6, which includes $\{(0,0), (1,0), (0,1), (2,0), (1,1)\}$, but does not include $(0,2)$. From this, we can easily see that $\Delta(I) = \{(0,0), (1,0), (0,1), (2,0), (1,1), (2,1)\}$, or $\Delta(I) = \{(0,0), (1,0), (0,1), (2,0), (1,1), (3,0)\}$. In the former case we call I an ideal of *type 62*, and in the latter case, we call I an ideal of *type 63*.

Lattice points surrounding $\Delta(I)$ are $\{(0,2), (3,0)\}$ for I of type 62, and $\{(0,2), (2,1), (4,0)\}$ for I of type 63. So, forms of their Groebner bases are as in Table 1. Note there should be a polynomial corresponding to the lattice point $(4,0)$ in Groebner base for I of type 63. However, the polynomial can be immediately obtained as $F - Yf$ with the defining polynomial F and the polynomial $f = Y^2 + a_5XY + a_4X^2 + a_3Y + a_2X + a_1$. So, we omit it.

Type 64

Next, suppose the first four monomials $1, X, Y, X^2$ are linearly independent, but the fifth monomial XY is a linear sum of them in $V = R/I$. That is, $\Delta(I)$ includes $\{(0,0), (1,0), (0,1), (2,0)\}$, but does not include $(1,1)$.

Then, if $\Delta(I)$ does not include $(0,2)$, we must have $\Delta(I) = \{(0,0), (1,0), (0,1), (2,0), (3,0), (4,0)\}$. However, by the assumption, I includes a polynomial $f = Y^2 + \dots$ with the leading monomial Y^2 , so I includes $Yf - F = -a_0X^4 + \dots$. This means $(4,0) \notin \Delta(I)$, a contradiction. Thus, we see that $\Delta(I)$ must include $(0,2)$, and $\Delta(I) = \{(0,0), (1,0), (0,1), (2,0), (0,2), (3,0)\}$. In this case, we call I an ideal of *type 64*.

Lattice points surrounding $\Delta(I)$ are $\{(0,3), (1,1), (4,0)\}$.

Hence, the form of Groebner base of I of type 64 is as in Table 1.

Type 65

Next suppose the first three monomials $1, X, Y$ are linearly independent, but the fourth monomial X^2 is a linear sum of them in $V = R/I$. Then, the ideal I include a polynomial f with the leading term X^2 . And we have $\Delta(I) = \{(0, 0), (1, 0), (0, 1), (1, 1), (0, 2), (1, 2)\}$. In this case, we call I an ideal of type 65. Since lattice points surrounding $\Delta(I)$ are $\{(0,3),(2,0)\}$, we know I is a principal ideal generated by f as in Table 1 (note the lattice point $(0,3)$ corresponds to the defining polynomial F).

A polynomial f with the leading term Y does not vanish on the six points corresponding to I , because $\deg(f)_0 = -v_{P_\infty}(f) = 4 < 6$. Hence, the first three monomials $1, X, Y$ are always linearly independent in $V = R/I$.

Now classification of ideals of order 6 is completed.

5.2 All ideal types of order not greater than 6

Ideals of order less than 6 are also similarly classified. We only show the result of classification in Table 1. Ideals of type 65,44 and 33 are principal ideals, units in Jacobian. Among all of the ideal types, only ideals of type 31,21,22 and 11 are reduced. For example, we can see that ideals of type 32 are not reduced as follows.

Let I be an ideal of type 32. Then $f_I = Y + a_2X + a_1$. So,

$$\delta(I^*) = -v_\infty(f_I) - \delta(I) = 4 - 3 = 1.$$

We know I^* is of type 11 and $f_{I^*} = X + a'_1$. So,

$$\delta(I^{**}) = -v_\infty(f_{I^*}) - \delta(I^*) = 3 - 1 = 2.$$

Since orders are distinct, $I \neq I^{**}$.

6. An Addition Algorithm in Jacobian of C_{34} curve — concrete level

Let $R = k[X, Y]/F$ be the coordinate ring of a C_{34} curve C defined by a polynomial F (Equation (1)) over a finite field k . In this section, we put Algorithm 1 into more concrete shape and estimate its efficiency. From now, aiming at an application for cryptography, we assume the order of the definition field k is large enough.

6.1 Composition 1

First, we deal with the first step of Algorithm 1 for distinct ideals I_1, I_2 . That is, we compute f_J for the ideal product $J = I_1 \cdot I_2$. For that sake, it is sufficient to find Groebner base of J with respect to C_{34} order

Table 1 All ideal types of order not greater than 6

| Order | Type | Form of Groebner base |
|-------|------|---|
| 6 | 61 | $\{X^3 + a_6Y^2 + a_5XY + a_4X^2 + a_3Y + a_2X + a_1, X^2Y + b_6Y^2 + b_5XY + b_4X^2 + b_3Y + b_2X + b_1, XY^2 + c_6Y^2 + c_5XY + c_4X^2 + c_3Y + c_2X + c_1\}$ |
| 6 | 62 | $\{Y^2 + a_5XY + a_4X^2 + a_3Y + a_2X + a_1, X^3 + b_5XY + b_4X^2 + b_3Y + b_2X + b_1\}$ |
| 6 | 63 | $\{Y^2 + a_5XY + a_4X^2 + a_3Y + a_2X + a_1, X^2Y + b_6X^3 + b_5XY + b_4X^2 + b_3Y + b_2X + b_1\}$ |
| 6 | 64 | $\{XY + a_4X^2 + a_3Y + a_2X + a_1, X^4 + b_6X^3 + b_5Y^2 + b_4X^2 + b_3Y + b_2X + b_1\}$ |
| 6 | 65 | $\{X^2 + a_3Y + a_2X + a_1\}$ |
| 5 | 51 | $\{Y^2 + a_5XY + a_4X^2 + a_3Y + a_2X + a_1, X^3 + b_5XY + b_4X^2 + b_3Y + b_2X + b_1, X^2Y + c_5XY + c_4X^2 + c_3Y + c_2X + c_1\}$ |
| 5 | 52 | $\{XY + a_4X^2 + a_3Y + a_2X + a_1, Y^2 + b_4X^2 + b_3Y + b_2X + b_1\}$ |
| 5 | 53 | $\{XY + a_4X^2 + a_3Y + a_2X + a_1, X^3 + b_5Y^2 + b_4X^2 + b_3Y + b_2X + b_1\}$ |
| 5 | 54 | $\{X^2 + a_3Y + a_2X + a_1, XY^2 + b_5Y^2 + b_4XY + b_3Y + b_2X + b_1\}$ |
| 4 | 41 | $\{XY + a_4X^2 + a_3Y + a_2X + a_1, Y^2 + b_4X^2 + b_3Y + b_2X + b_1, X^3 + c_4X^2 + c_3Y + c_2X + c_1\}$ |
| 4 | 42 | $\{X^2 + a_3Y + a_2X + a_1, XY + b_3Y + b_2X + b_1\}$ |
| 4 | 43 | $\{X^2 + a_3Y + a_2X + a_1, Y^2 + b_4XY + b_3Y + b_2X + b_1\}$ |
| 4 | 44 | $\{Y + a_2X + a_1\}$ |
| 3 | 31 | $\{X^2 + a_3Y + a_2X + a_1, XY + b_3Y + b_2X + b_1, Y^2 + c_3Y + c_2X + c_1\}$ |
| 3 | 32 | $\{Y + a_2X + a_1, X^3 + b_3X^2 + b_2X + b_1\}$ |
| 3 | 33 | $\{X + a_1\}$ |
| 2 | 21 | $\{Y + a_2X + a_1, X^2 + b_2X + b_1\}$ |
| 2 | 22 | $\{X + a_1, Y^2 + b_2Y + b_1\}$ |
| 1 | 11 | $\{X + a_1, Y + b_1\}$ |

(f_J is the first element of it).

Since the genus of C_{34} curves is three, types of input ideals for Algorithm 1 are either 11,21,22,31 or 32. Here, we only discuss the case in which ideals I_1, I_2 are both of type 31. Another cases are dealt with similarly.

Suppose we choose distinct ideals I_1, I_2 of type 31 at random from Jacobian group. Then since we assume the order q of k is large enough, almost always (with the probability approximately $(q - 1)/q$) we have

$$V(I_1) \cap V(I_2) = \emptyset \tag{2}$$

where $V(I)$ denotes the zero set of an ideal I . So, first we assume the condition (2).

Let $J = I_1I_2$ be the ideal product of I_1 and I_2 . Since the order of J is 6, the type of J is either 61,62,63,64 or 65. To determine which it is, by Table 1, we see it is sufficient to find linear relations among 10 monomials

$$1, X, Y, X^2, XY, Y^2, X^3, X^2Y, XY^2, X^4 \tag{3}$$

in the vector space R/J over k .

Since I_i ($i = 1, 2$) is of type 31, we have

$$R/I_i \simeq k \cdot 1 \oplus k \cdot X \oplus k \cdot Y. \quad (4)$$

$$m \mapsto v_m^{(i)} \quad (5)$$

By condition (2), we have

$$R/J \simeq R/I_1 \oplus R/I_2 \simeq \bigoplus_{i=1}^6 k, \quad (6)$$

$$m \mapsto (m \bmod (I_1), m \bmod (I_2)) \mapsto v_m^{(1)} : v_m^{(2)}$$

where $v_m^{(1)} : v_m^{(2)}$ denotes the sixth dimensional vector over k obtained by connecting two vectors $v_m^{(i)}$ ($i = 1, 2$).

Thus, to obtain linear relations in R/J among 10 monomials m_i in equation (3), it is sufficient to find linear relations among rows of the 10×6 matrix M_C which is obtained by lining up vectors $v_{m_i}^{(1)} : v_{m_i}^{(2)}$ ($i = 1, 2, \dots, 10$)

Linear relations among rows of M_C can be obtained by making M_C triangular using the row reduce procedure as well known, and we get the type of ideal J and its Groebner base. More details are shown through the following example.

When condition (2) does not hold for ideals I_1, I_2 , the rank M_C becomes less than 6. After making M_C triangular, if we know the rank of M_C is less than 6, then we generate R_i satisfying $R_1 + R_2 = 0$, and compute $(I_1 + R_1) + (I_2 + R_2)$ instead of $I_1 + I_2$. Here, '+' denotes the addition in Jacobian.

6.1.1 Example

For example, we deal with C_{34} curve $Y^3 + X^4 + 7X = 0$ on the prime field of characteristics $p = 1009$. Take the following two ideals of type 31:

$$I_1 = \{X^2 + 726Y + 836X + 355, XY + 36Y + 428X + 477, Y^2 + 746Y + 425X + 865\}$$

$$I_2 = \{X^2 + 838Y + 784X + 97, XY + 602Y + 450X + 291, Y^2 + 506Y + 524X + 497\}$$

We would like to compute Groebner base of $J = I_1 I_2$ to find f_J . By computing the remainder of each m_i in equation (3) modulo I_1 and I_2 respectively, we get the matrix M_C for I_1, I_2 :

$$M_C = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 654 & 173 & 283 & 912 & 225 & 171 \\ 532 & 581 & 973 & 718 & 559 & 407 \\ 144 & 584 & 263 & 512 & 485 & 503 \\ 349 & 269 & 429 & 53 & 821 & 109 \\ 609 & 418 & 243 & 888 & 856 & 916 \\ 199 & 720 & 418 & 310 & 331 & 91 \\ 554 & 498 & 143 & 643 & 522 & 107 \end{pmatrix}.$$

To obtain linear relations among rows of M_C , we connect M_C and 10-th unit matrix I_{10} to get $M'_C =$

$M_C : I_{10}$. Against M'_C , we apply the row reduce procedure up to the sixth row and we obtain $m =$

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 258 & 52 & 897 & 355 & 836 & 726 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 621 & 688 & 268 & 365 & 592 & 187 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 31 & 514 & 469 & 637 & 669 & 155 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 28 & 132 & 31 & 271 & 469 & 166 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 856 & 618 & 747 & 909 & 132 & 636 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 652 & 322 & 240 & 978 & 826 & 846 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 333 & 346 & 980 & 935 & 824 & 614 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

The result shows the first six rows of M_C are linearly independent. This means monomials $1, X, Y, X^2, XY, Y^2$ are linearly independent in R/J and the product J is of type 61.

Moreover, the right 10 elements of the seventh, eighth and ninth rows of m represents liner expressions of the seventh, eighth and ninth rows of M_C by the first six rows of M_C , respectively. From this, we know linear expressions of X^3, X^2Y, XY^2 by $1, X, Y, X^2, XY, Y^2$ in R/J , respectively, and we get the following Groebner base of J :

$$J = \{28 + 132X + 31Y + 271X^2 + 469XY + 166Y^2 + X^3, \\ 856 + 618X + 747Y + 909X^2 + 132XY + 636Y^2 + X^2Y, \\ 652 + 322X + 240Y + 978X^2 + 826XY + 846Y^2 + XY^2\}.$$

Hence, we have $f_J = 28 + 132X + 31Y + 271X^2 + 469XY + 166Y^2 + X^3$.

6.2 Composition 2

We see that the first step of Algorithm 1 for the same two ideals $I_1 = I, I_2 = I$ in R can be performed in the similar way as the first step for distinct two ideals.

We want to compute Groebner base of the ideal product $J = I^2$ to get f_J . As in section 6.1, we only deal with an ideal I of type 31. Other cases are handled similarly.

Since we assume the order q of k is large enough, almost always (with the probability approximately $(q-1)/q$) it holds that

$$V(I) \text{ has no multiple point.} \quad (7)$$

So, first we assume the condition (7).

Since the order of $J = I^2$ is also 6, it is sufficient to find linear relations in R/J among monomials in equation (3). By condition (7), the necessary and sufficient condition to $f \in R$ belongs to $J = I^2$ is

$$f \in I, f_X F_Y - f_Y F_X \in I.$$

So, we have

$$R/J \simeq R/I \oplus R/I \simeq \bigoplus_{i=1}^6 k \quad (8)$$

$$m \mapsto (m, m_X F_Y - m_Y F_X) \mapsto v_m : v_{(m_X F_Y - m_Y F_X)}$$

where $v_m : v_{(m_X F_Y - m_Y F_X)}$ is a sixth dimensional vector over k obtained by connecting two vectors $v_m, v_{(m_X F_Y - m_Y F_X)}$.

Thus, to obtain linear relations in R/J among

m_i in equation (3), it is sufficient to find linear relations among rows of 10×6 matrix M_D which is obtained by lining up vectors $v_{m_i} : v_{(m_i X F_Y - m_i Y F_X)}$ ($i = 1, 2, \dots, 10$).

Just as in section 6.1, we make M_D triangular by the row reduce procedure to obtain the type of J and its Groebner base.

If condition (7) does not hold for I , the rank of M_D is less than 6. After making M_D triangular, if we know the rank of M_D is less than 6, then we generate R_i satisfying $R_1 + R_2 = 0$, and compute $(I + R_1) + (I + R_2)$ instead of $I + I$. Here, '+' denotes the addition in Jacobian.

6.3 Reduction

We consider the second (and the third) step of Algorithm 1. That is, we compute Groebner base of $J^* = f_J : J$ for an ideal J of order not greater than 6. Here we only deal with J of type 61. Other types of J are dealt with similarly.

Since J is of type 61, J can be written as

$$\{f_J = X^3 + a_6 Y^2 + \dots, g = X^2 Y + b_6 Y^2 + \dots, h = X Y^2 + c_6 Y^2 + \dots\}.$$

Since $J^* = f_J : J$ from definition, we have $\delta(J^*) = -v_\infty(f_J) - \delta(J) = 3$. Moreover J^* is reduced by Proposition 1, so the type of J^* must be 31 (see Remark in section 5).

Hence, to find Groebner base of J^* , it is sufficient to find linear relations $\sum_i d_i m_i$ for m_i in

$$1, X, Y, X^2, XY, Y^2 \tag{9}$$

such that both $\sum_i d_i m_i g$ and $\sum_i d_i m_i h$ are equal to 0 in R/f_J .

Since $LM(F) = Y^3$, $LM(f_J) = X^3$, we have

$$\begin{aligned} R/f_J R &\simeq k \cdot 1 \oplus k \cdot X \oplus k \cdot Y \oplus k \cdot X^2 \oplus k \cdot XY \\ &\oplus k \cdot Y^2 \oplus k \cdot X^2 Y \oplus k \cdot XY^2 \oplus k \cdot X^2 Y^2. \\ f &\mapsto w_f \end{aligned} \tag{10}$$

So, to find those linear relations among m_i in equation (9), it is sufficient to find linear relations among rows of 6×18 matrix M_R which is obtained by lining up vectors $w_{m_i g} : w_{m_i h}$ ($i = 1, 2, \dots, 6$).

Just as in section 6.1, we make M_R triangular by the row reduce procedure to obtain the type of J^* and its Groebner base.

6.3.1 Example 2

For example, we deal with C_{34} curve $Y^3 + X^4 + 7X = 0$ on the prime field of characteristics $p = 1009$, again. Take the following ideal of type 61, which was produced by Composition 1 in Section 6.1.1:

$$J = \{f_J = 28 + 132X + 31Y + 271X^2 + 469XY +$$

$$\begin{aligned} &166Y^2 + X^3, \\ &g = 856 + 618X + 747Y + 909X^2 + 132XY + \\ &636Y^2 + X^2Y, \\ &h = 652 + 322X + 240Y + 978X^2 + 826XY + \\ &846Y^2 + XY^2\} \end{aligned}$$

We would like to compute Groebner base of the quotient ideal $J^* = f_J : J$. Since J is of type 61, $R/f_J R$ is given by Equation (10) and the 6×18 matrix M_R is obtained by arranging vectors $w_{m_i g} : w_{m_i h}$ ($i = 1, 2, \dots, 6$) in rows:

$$M_R = \begin{pmatrix} 856 & 618 & 747 & 909 & \dots & 240 & 978 & 826 & 846 & 0 & 1 & 0 \\ 149 & 667 & 173 & 220 & \dots & 961 & 651 & 653 & 101 & 826 & 846 & 1 \\ 79 & 179 & 475 & 357 & \dots & 590 & 358 & 694 & 31 & 473 & 939 & 166 \\ 102 & 241 & 513 & 394 & \dots & 809 & 560 & 425 & 535 & 552 & 671 & 763 \\ 367 & 1 & 403 & 54 & \dots & 851 & 289 & 210 & 802 & 321 & 522 & 278 \\ 944 & 384 & 763 & 956 & \dots & 415 & 643 & 539 & 438 & 887 & 102 & 363 \end{pmatrix}.$$

To obtain linear relations among rows of M_R , we connect M_R and 6-th unit matrix I_6 to get $M'_R = M_R : I_6$. Against M'_R , we apply the row reduce procedure up to the third row and obtain:

$$m = \begin{pmatrix} 856 & 618 & 747 & \dots & 846 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 62 & 485 & \dots & 272 & 826 & 814 & 1 & 977 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 252 & \dots & 748 & 221 & 979 & 217 & 281 & 51 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 982 & 226 & 146 & 1 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 449 & 79 & 320 & 0 & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 544 & 564 & 195 & 0 & 0 & 1 \end{pmatrix}.$$

The result shows the first three rows of M_R are linearly independent. So, the quotient J^* must be of type 31. Moreover, the right six elements of the fourth, fifth and sixth rows of m gives us the coefficients in the Groebner base of J^* of type 31:

$$\begin{aligned} J^* &= \{982 + 226X + 146Y + X^2, \\ &449 + 79X + 320Y + XY, \\ &544 + 564X + 195Y + Y^2\}. \end{aligned}$$

6.4 A pseudo-code of the addition algorithm in Jacobian of C_{34} curves

Now, by the above discussion, we finally get a following pseudo-code of the algorithm of addition in Jacobian of C_{34} curve. In the following, for saving space, only codes involving ideals of type 61 and 31 are shown. Other parts of the code are driven in the same manner.

```

algorithm JSum
inputs  $I_1$  : ideal,  $I_2$  : ideal,
an output  $J^*$  : ideal
IF type( $I_1$ ) == 65 or 44 or 33 THEN
  RETURN  $I_2$ 
IF type( $I_2$ ) == 65 or 44 or 33 THEN
  RETURN  $I_1$ 
IF  $I_1 \neq I_2$  THEN  $J \leftarrow$  Compose1( $I_1, I_2$ )
ELSE  $J \leftarrow$  Compose2( $I_1$ )
IF  $J$  == 'error' THEN
   $R_1 \leftarrow$  a random element in Jacobian,
   $R_2 \leftarrow$  Reduce( $R_1$ )
  RETURN JSum(JSum( $I_1, R_1$ ),
    JSum( $I_2, R_2$ ))
IF type( $J$ ) == 65 or 44 or 33 THEN
  RETURN  $J$ 
  
```

$J^* \leftarrow \text{Reduce}(J)$
 $J^{**} \leftarrow \text{Reduce}(J^*)$
RETURN J^{**}

algorithm Compose1

inputs I_1 : ideal, I_2 : ideal
 an output J : ideal

IF $\text{type}(I_1) == 31$ **AND** $\text{type}(I_2) == 31$ **THEN**

$$M_C \leftarrow \begin{pmatrix} v_1^{(1)} : v_1^{(2)} \\ v_X^{(1)} : v_X^{(2)} \\ v_Y^{(1)} : v_Y^{(2)} \\ v_{X^2}^{(1)} : v_{X^2}^{(2)} \\ v_{XY}^{(1)} : v_{XY}^{(2)} \\ v_{Y^2}^{(1)} : v_{Y^2}^{(2)} \\ v_{X^3}^{(1)} : v_{X^3}^{(2)} \\ v_{X^2Y}^{(1)} : v_{X^2Y}^{(2)} \\ v_{XY^2}^{(1)} : v_{XY^2}^{(2)} \\ v_{X^4}^{(1)} : v_{X^4}^{(2)} \end{pmatrix}$$

$J \leftarrow \text{GetGB}(6, M_C)$

ELSE IF

/ omitted */*

RETURN J

algorithm Compose2

inputs I : ideal, output J : ideal

IF $\text{type}(I) == 31$ **THEN**

$$M_D \leftarrow \begin{pmatrix} v_1 : 0 \\ v_X : v(F_Y) \\ v_Y : v(-F_X) \\ v_{X^2} : v(2F_Y X) \\ v_{XY} : v(-F_X X + F_Y Y) \\ v_{Y^2} : v(-2F_X Y) \\ v_{X^3} : v(3F_Y X^2) \\ v_{X^2Y} : v(-F_X X^2 + 2F_Y XY) \\ v_{XY^2} : v(-2F_X XY + F_Y Y^2) \\ v_{X^4} : v(4F_Y X^3) \end{pmatrix}$$

$J \leftarrow \text{GetGB}(6, M_D)$

ELSE IF

/ omitted */*

RETURN J

algorithm GetGB

inputs d : integer, M : matrix

an output J : ideal

IF $d == 6$ **THEN**

$M' \leftarrow M : I_6$

$m \leftarrow \text{RowReduce}(M', 6)$

m_i denotes the i -th row of the matrix m .

IF $m_1, m_2, m_3, m_4, m_5, m_6$ are l. indep.,

THEN $\text{type}(J) \leftarrow 61$

$J \leftarrow \{m_{7,7} + m_{7,8}X + m_{7,9}Y + m_{7,10}X^2$

$+ m_{7,11}XY + m_{7,12}Y^2 + X^3,$

$m_{8,7} + m_{8,8}X + m_{8,9}Y + m_{8,10}X^2$

$+ m_{8,11}XY + m_{8,12}Y^2 + X^2Y,$

$m_{9,7} + m_{9,8}X + m_{9,9}Y + m_{9,10}X^2$

$+ m_{9,11}XY + m_{9,12}Y^2 + XY^2\}$

ELIF $m_1, m_2, m_3, m_4, m_5, m_8$ are l. indep.,

THEN $\text{type}(J) \leftarrow 62$

$J \leftarrow \{m_{6,7} + m_{6,8}X + m_{6,9}Y + m_{6,10}X^2$

$+ m_{6,11}XY + Y^2,$

$m_{7,7} + m_{7,8}X + m_{7,9}Y + m_{7,10}X^2$

$+ m_{7,11}XY + X^3\}$

ELIF $m_1, m_2, m_3, m_4, m_5, m_7$ are l. indep.,

THEN $\text{type}(J) \leftarrow 63$

$J \leftarrow \{m_{6,7} + m_{6,8}X + m_{6,9}Y + m_{6,10}X^2$

$+ m_{6,11}XY + Y^2,$

$m_{8,7} + m_{8,8}X + m_{8,9}Y + m_{8,10}X^2$

$+ m_{8,11}XY + m_{8,13}X^3 + X^2Y\}$

ELIF $m_1, m_2, m_3, m_4, m_6, m_7$ are l. indep.,

THEN $\text{type}(J) \leftarrow 64$

$J \leftarrow \{m_{5,7} + m_{5,8}X + m_{5,9}Y + m_{5,10}X^2$

$+ XY,$

$m_{10,7} + m_{10,8}X + m_{10,9}Y + m_{10,10}X^2$

$+ m_{10,12}Y^2 + m_{10,13}X^3 + X^4\}$

ELIF $m_1, m_2, m_3, m_5, m_6, m_9$ are l. indep.,

THEN $\text{type}(J) \leftarrow 65$

$J \leftarrow \{m_{4,7} + m_{4,8}X + m_{4,9}Y + X^2\}$

ELSE $J \leftarrow \text{'error'}$

ELSE IF

/ omitted */*

RETURN J

algorithm Reduce

an input J : ideal

an output J^* : ideal

IF $\text{type}(J) == 61$ **THEN**

$f = X^3 + a_6Y^2 + a_5XY + a_4X^2$

$+ a_3Y + a_2X + a_1$

\leftarrow the first element of J

$g = X^2Y + b_6Y^2 + b_5XY + b_4X^2$

$+ b_3Y + b_2X + b_1$

\leftarrow the second element of J

$h = XY^2 + c_6Y^2 + c_5XY + c_4X^2$

$+ c_3Y + c_2X + c_1$

\leftarrow the third element of J

IF $(-a_5 - a_6^2 + b_6) \neq 0$ **THEN**

$$M_r \leftarrow \begin{pmatrix} 1 & 0 & 0 \\ -a_4 - a_5a_6 + b_5 & -a_5 - a_6^2 + b_6 & 1 \\ b_4 + a_5b_6 & b_5 + a_6b_6 & -a_5 - a_6^2 + b_6 \\ e_{4,1} & e_{4,2} & e_{5,3} \\ e_{5,1} & e_{5,2} & e_{6,3} \\ e_{6,1} & e_{6,2} & e_{6,3} \end{pmatrix}$$

For definitions of $e_{i,j}$, see Equation (11).

$M'_r \leftarrow M_r : I_3$

$m \leftarrow \text{RowReduce}(M'_r, 3)$

$\text{type}(J^*) \leftarrow 31$

$J^* \leftarrow \{m_{4,4} + m_{4,5}X + m_{4,6}Y + X^2,$

$m_{5,4} + m_{5,5}X + m_{5,6}Y + XY,$

$m_{6,4} + m_{6,5}X + m_{6,6}Y + Y^2\}$

ELSE

$$M_R \leftarrow \begin{pmatrix} w_g : w_h \\ w_{Xg} : w_{Xh} \\ w_{Yg} : w_{Yh} \\ w_{X^2g} : w_{X^2h} \\ w_{XYg} : w_{XYh} \\ w_{Y^2g} : w_{Y^2h} \end{pmatrix}$$

$M'_R \leftarrow M_R : I_6$

$m \leftarrow \text{RowReduce}(M'_R, 3)$

$\text{type}(J^*) \leftarrow 31$

$J^* \leftarrow \{m_{4,19} + m_{4,20}X + m_{4,21}Y + X^2,$

$m_{5,19} + m_{5,20}X + m_{5,21}Y + XY,$

$m_{6,19} + m_{6,20}X + m_{6,21}Y + Y^2\}$

ELSE IF

/ omitted */*

ELSE IF $\text{type}(J) == 31$ **THEN**

$f = X^2 + a_3Y + a_2X + a_1$

\leftarrow the first element of J

$g = XY + b_3Y + b_2X + b_1$

\leftarrow the second element of J

$h = Y^2 + c_3Y + c_2X + c_1$

\leftarrow the third element of J

IF $a_3 \neq 0$ **THEN**

$M_r \leftarrow$

$$\begin{pmatrix} 1 & 0 & 0 \\ -a_2 + b_3 & -a_3 & 0 \\ b_2 & b_3 & 1 \\ f_{4,1} & f_{4,2} & -a_2 + b_3 \\ f_{5,1} & f_{5,2} & -a_3^2 + b_2 \end{pmatrix}$$

$f_{4,1} = 2a_2a_3^2 + b_1 - a_2b_2$

$f_{4,2} = -a_1 + a_3^3 - a_3b_2$

$f_{5,1} = -2a_1a_3 + 3a_3^2a_3 - 2a_2a_3b_3$

$f_{5,2} = 2a_2a_3^2 + b_1 - a_3^2b_3$

```

M'_r ← M_r : I_3
m ← RowReduce(M'_r, 3)
type(J*) ← 31
J* ← {a_1 + a_2X + a_3Y + X^2,
      m_{4,4} + m_{4,5}X + m_{4,6}Y + XY,
      m_{5,4} + m_{5,5}X + m_{5,6}Y + Y^2}
    
```

ELSE

$$M_R \leftarrow \begin{pmatrix} w_g : w_h \\ w_{Xg} : w_{Xh} \\ w_{Yg} : w_{Yh} \\ w_{XYg} : w_{XYh} \\ w_{Y^2g} : w_{Y^2h} \end{pmatrix}$$

```

M'_R ← M_R : I_5
m ← RowReduce(M'_R, 3)
type(J*) ← 31
J* ← {the first element of J
      m_{4,13} + m_{4,14}X + m_{4,15}Y + XY,
      m_{5,13} + m_{5,14}X + m_{5,15}Y + Y^2}
    
```

ELSE IF

/* omitted */

RETURN J*

algorithm RowReduce

an input M : matrix

an output d : integer

#Until independent d rows are obtained,

#repeat row reduce procedure.

$n \leftarrow$ the row number of M , $b \leftarrow$ the column number of M
 $dim \leftarrow 0$, $i \leftarrow 1$

WHILE $dim < d$ **AND** $i \leq n$ **DO**

IF $M_{i,dim+1} == 0$ **THEN**

$k \leftarrow dim + 2$

WHILE $M_{i,k} == 0$ **AND** $k \leq b - n$ **DO** $k \leftarrow k + 1$

IF $k \leq b - n$ **THEN** Exchange $dim + 1$ -th and k -th
 columns of M .

$c \leftarrow M_{i,dim+1}$

IF $c == 0$ **THEN** $i \leftarrow i + 1$ **NEXT**

$dim \leftarrow dim + 1$

$c \leftarrow c^{-1}$

FOR $j \leftarrow i + 1, \dots, n$ **DO**

M_i denotes the i -th row of the matrix M .

$M_j \leftarrow M_j - c \cdot M_{j,dim} \cdot M_i$

$i \leftarrow i + 1$

RETURN M

7. Estimation of the performance

We make an estimation of the performance of the pseudo-code in Section 6.4. Let q be the order of the definition field k . A random element in Jacobian is represented by an ideal of type 31 with the probability about $(q-1)/q$. Also, outputs of Compose 1,2 for ideals of type 31 are ideals of type 61 with the probability about $(q-1)/q$. So, to estimate the efficiency of the pseudo-code, it is sufficient to estimate the amount of computation of Compose1, 2 for ideals of type 31 and the amount of computation of Reduce for ideals of type 61 and 31. In the following, we describe the amount of computation by the number of times of multiplication and inverse of elements in k .

First, we see the amount of computation of Compose1. Let I_1, I_2 be ideals of type 31:

$$I_1 = \{X^2 + a_3Y + a_2X + a_1,$$

$$XY + b_3Y + b_2X + b_1, Y^2 + c_3Y + c_2X + c_1\},$$

$$I_2 = \{X^2 + s_3Y + s_2X + s_1,$$

$$XY + t_3Y + t_2X + t_1, Y^2 + u_3Y + u_2X + u_1\}.$$

For ideals I_1, I_2 , the matrix M_C is represented as

$$M_C = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ -a_1 & -a_2 & -a_3 & -s_1 & -s_2 & -s_3 \\ -b_1 & -b_2 & -b_3 & -t_1 & -t_2 & -t_3 \\ -c_1 & -c_2 & -c_3 & -u_1 & -u_2 & -u_3 \\ e_{7,1} & e_{7,2} & e_{7,3} & e_{7,4} & e_{7,5} & e_{7,6} \\ e_{8,1} & e_{8,2} & e_{8,3} & e_{8,4} & e_{8,5} & e_{8,6} \\ e_{9,1} & e_{9,2} & e_{9,3} & e_{9,4} & e_{9,5} & e_{9,6} \\ e_{10,1} & e_{10,2} & e_{10,3} & e_{10,4} & e_{10,5} & e_{10,6} \end{pmatrix}$$

$$e_{7,1} = a_1a_2 + a_3b_1, e_{7,2} = -a_1 + a_2^2 + a_3b_2, e_{7,3} = a_2a_3 + a_3b_3,$$

$$e_{7,4} = s_1s_2 + s_3t_1, e_{7,5} = -s_1 + s_2^2 + s_3t_2, e_{7,6} = s_2s_3 + s_3t_3,$$

$$e_{8,1} = a_2b_1 + a_3c_1, e_{8,2} = a_2b_2 + a_3c_2, e_{8,3} = -a_1 + a_2b_3 + a_3c_3,$$

$$e_{8,4} = s_2t_1 + s_3u_1, e_{8,5} = s_2t_2 + s_3u_2, e_{8,6} = -s_1 + s_2t_3 + s_3u_3,$$

$$e_{9,1} = b_1b_2 + b_3c_1, e_{9,2} = b_2^2 + b_3c_2, e_{9,3} = -b_1 + b_2b_3 + b_3c_3,$$

$$e_{9,4} = t_1t_2 + t_3u_1, e_{9,5} = t_2^2 + t_3u_2, e_{9,6} = -t_1 + t_2t_3 + t_3u_3,$$

$$e_{10,1} = a_1^2 - a_1a_2^2 - 2a_2a_3b_1 - a_3^2c_1,$$

$$e_{10,2} = 2a_1a_2 - a_3^2 - 2a_2a_3b_2 - a_3^2c_2,$$

$$e_{10,3} = 2a_1a_3 - a_2^2a_3 - 2a_2a_3b_3 - a_3^2c_3,$$

$$e_{10,4} = s_1^2 - s_1s_2^2 - 2s_2s_3t_1 - s_3^2u_1,$$

$$e_{10,5} = 2s_1s_2 - s_3^2 - 2s_2s_3t_2 - s_3^2u_2,$$

$$e_{10,6} = 2s_1s_3 - s_2^2s_3 - 2s_2s_3t_3 - s_3^2u_3.$$

From this representation, we see the matrix M_C can be constructed in at most 44 multiplications, removing duplication adequately. Knowing the first three rows of M'_C are already row-reduced, and elements of them are 0 or 1, and assuming the output ideal would be of type 61, we see RowReduce for M'_C can be performed in 3 inverses and at most $6 \cdot 6 + 6 \cdot 5 + 6 \cdot 4 = 90$ times multiplications. Thus, Compose1 are performed in at most 3 inverses and 134 multiplications.

Similarly, we can see Compose2 are performed in at most 3 inverses and 214 multiplications. As M_D is more complicated than M_C , times of multiplication is increased.

Next we estimate the amount of computation of Reduce for an ideal of type 61. Let J be an ideal of type 61:

$$J = \{X^3 + a_6Y^2 + a_5XY + a_4X^2 + a_3Y + a_2X + a_1,$$

$$X^2Y + b_6Y^2 + b_5XY + b_4X^2 + b_3Y + b_2X + b_1,$$

$$XY^2 + c_6Y^2 + c_5XY + c_4X^2 + c_3Y + c_2X + c_1\}$$

The 6×3 sub-matrix M_r , obtained by extracting the seventh, eighth and ninth columns of M_R for J , is represented as

$$M_r = \begin{pmatrix} 1 & 0 & 0 \\ -a_4 - a_5a_6 + b_5 & -a_5 - a_6^2 + b_6 & 0 \\ b_4 + a_5b_6 & b_5 + a_6b_6 & 1 \\ e_{4,1} & e_{4,2} & -a_5 - a_6^2 + b_6 \\ e_{5,1} & e_{5,2} & e_{5,3} \\ e_{6,1} & e_{6,2} & e_{6,3} \end{pmatrix} \quad (11)$$

$$e_{4,1} = -a_2 + a_4^2 - a_3a_6 + 3a_4a_5a_6 + a_5^2a_6^2 + b_3 - a_5b_4 - a_4b_5 - a_5a_6b_5,$$

$$e_{4,2} = -a_3 + a_4a_5 + a_5^2a_6 + 2a_4a_6^2 + a_5a_6^3 - a_6b_4 - a_5b_5 - a_6^2b_5,$$

$$e_{5,1} = -2a_3a_5 + 2a_4a_5^2 - a_2a_6 + a_4^2a_6 + a_5^3a_6 - a_3a_6^2 + 3a_4a_5a_6^2 +$$

$$a_5^2a_6^3 + b_2 - a_4b_4 - a_5a_6b_4 + a_3b_6 - 2a_4a_5b_6 - a_5^2a_6b_6,$$

$$e_{5,2} = -a_2 + a_5^3 - 2a_3a_6 + 2a_4a_5a_6 + 2a_5^2a_6^2 + 2a_4a_6^3 + a_5a_6^4 +$$

Table 2 Amount of computation of the pseudo-code in Section 6.4

| | Addition | Doubling |
|------------------------|----------|----------|
| Compose | 134M+3I | 214M+3I |
| Reduce for the type 61 | 54M+I | 54M+I |
| Reduce for the type 31 | 16M+I | 16M+I |
| Total | 204M+5I | 284M+5I |

$$\begin{aligned}
& b_3 - a_5 b_4 - a_6^2 b_4 - a_5^2 b_6 - a_4 a_6 b_6 - a_5 a_6^2 b_6, \\
e_{5,3} &= -a_4 - 2a_5 a_6 - a_6^3 + b_5 + a_6 b_6, \\
e_{6,1} &= -2a_3 a_4 - 2a_2 a_5 + 3a_4^2 a_5 - 4a_3 a_5 a_6 + 6a_4 a_5^2 a_6 - a_2 a_6^2 + \\
& a_4^2 a_6^2 + 2a_5^3 a_6^2 - a_3 a_6^3 + 3a_4 a_5 a_6^3 + a_5^2 a_6^4 + a_5 b_3 + a_3 b_5 - \\
& 2a_4 a_5 b_5 - a_5^2 a_6 b_5 + a_2 b_6 - a_4^2 b_6 + a_3 a_6 b_6 - 3a_4 a_5 a_6 b_6 - a_5^2 a_6^2 b_6, \\
e_{6,2} &= -2a_3 a_5 + 2a_4 a_5^2 - 2a_2 a_6 + a_4^2 a_6 + 2a_5^3 a_6 - 3a_3 a_6^2 + \\
& 5a_4 a_5 a_6^2 + 3a_5^2 a_6^3 + 2a_4 a_6^4 + a_5 a_6^5 + b_2 + a_6 b_3 - a_5^2 b_5 - \\
& a_4 a_6 b_5 - a_5 a_6^2 b_5 + a_3 b_6 - a_4 a_5 b_6 - a_5^2 a_6 b_6 - 2a_4 a_6^2 b_6 - a_5 a_6^3 b_6, \\
e_{6,3} &= -a_5^2 - 2a_4 a_6 - 3a_5 a_6^2 - a_6^4 + b_4 + a_6 b_5 + a_5 b_6 + a_6^2 b_6.
\end{aligned}$$

Using this representation we know that if the $(2, 2)$ -element $d = -a_5 - a_6^2 + b_6$ of M_r is not equal to zero, the rank of M_r must be 3. So, if $d \neq 0$, we can use 6×3 matrix M_r instead of 6×18 matrix M_R . As the probability of $d = 0$ is about $1/q$, we can assume $d \neq 0$ to estimate the efficiency of the pseudo-code.

By equation (11), we see that the matrix M_r can be constructed in at most 40 multiplications, removing duplication adequately. Knowing the first three rows of M_r has the triangular form and its $(1, 1)$ and $(3, 3)$ elements are 1, we see RowReduce for M_r' can be performed in 1 inverse and at most $2 \cdot 4 + 2 \cdot 3 = 14$ times multiplications. Thus, Reduce for an ideal of type 61 can be performed in at most 1 inverses and 54 multiplications. Similarly, we can see that Reduce for an ideal of type 31 can be performed in at most 1 inverses and 16 multiplications.

Summarizing the above discussion, the amount of computation of the pseudo-code in Section 6.4 is given in the following Table 2. In the table, I and M denotes the operation of inverse and multiplication of elements in k , respectively.

We can add two points on an elliptic curve with one inverse and three multiplications of elements in the definition field, and can double a point with one inverse and four multiplications. Note to obtain the same size of Jacobian, elliptic curves require the definition field of 3 times of bits length of the one for C_{34} curve. Assuming the amount of computation of one inverse is equal to the amount of 10 times multiplication, and assuming the amount of computation of inverse or multiplication grows in the order of square of bit lengths, the amount of computation of the addition on C_{34} curve is $254/(13 \times 9) \approx 2.17$ times of the one for an elliptic curve, and the one of the double is $334/(14 \times 9) \approx 2.65$ times of the one for an elliptic curve.

Finally, for comparison, we see recent works for addition algorithm in genus three hyperelliptic curves in Table 3. In the table, I, M and A denotes the operation

Table 3 Results of addition algorithms in genus three hyperelliptic curves

| | Addition | Doubling |
|-------------------|-------------------|-------------------|
| Kuroki et al.[10] | 81M + I (+ 125 A) | 74M + I (+ 125A) |
| Pelzl et al.[15] | 76M + I (+ 95A) | 75M + I (+ 97A) |
| Gonda et al.[6] | 79 M + I (+ 83 A) | 78 M + I (+ 83 A) |

of inverse, multiplication and addition of elements in the definition field, respectively.

As seen, our addition algorithm is still very slower than those. However, note that our target, C_{34} curve, is non-hyperelliptic of genus three, and our method has made it possible to eliminate the use of Buchberger algorithm in the computation, just to use row reduction method in matrices. It is likely that our addition algorithm will be made much faster by employing similar techniques, such as composition through Chinese Remainder Theorem or Newton iteration and etc, used in the previous works of hyperelliptic case.

8. Conclusion

The paper gave an efficient algorithm to compute addition in Jacobian of C_{34} curves, aiming at C_{34} curve cryptosystems. C_{34} curve cryptosystems has two advantages. The first is safety. Since C_{34} curves have different mathematical features from elliptic or hyperelliptic curves, even if we encountered some new attack method which seriously damages of safety of elliptic or hyperelliptic curve cryptosystems, C_{34} curves possibly would survive. The second advantage is the short size of the base field. If we use 64-bit CPU, C_{34} curve cryptosystems don't require multi-precision package.

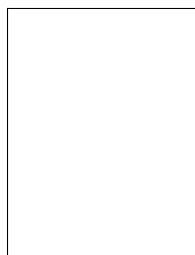
In the paper, we modified the addition algorithm of [1] for C_{ab} curves in the specific manner to C_{34} curves. We classified all of the forms of the Groebner bases of ideals involved in the addition in Jacobian of C_{34} curve and we saw that the classification eliminated the use of Buchberger algorithm from the algorithm of [1]. We showed that our resulting algorithm computes the addition in Jacobian of C_{34} curves in about 3 times amount of computation of the one in elliptic curves, when the sizes of groups are set to be the same.

Finally, we compared the efficiency of our algorithm with recent works for addition algorithm in genus three hyperelliptic curves. Unfortunately, our addition algorithm is still very slower than those. However, it is likely that our addition algorithm will be made much faster by employing similar techniques used in the previous works of hyperelliptic case. Our result gives the first step toward a practical C_{34} curve cryptosystem.

References

- [1] S. Arita, "Algorithms for computations in Jacobian group of C_{ab} curve and their application to discrete-log-based public key cryptosystems," IEICE TRANS. FOUND., VOL.J82-A, NO.8, pp.1291-1299, 1999.

- [2] S. Arita, "An Addition Algorithm in Jacobian of C_{34} Curve," ACISP 2003, LNCS 2727, pp.93–105, Wollongong, Australia, 2003.
- [3] D.G.Cantor, "Computing in the Jacobian of a hyperelliptic curve", Mathematics of Computation, 48(177), pp.95-101,1987.
- [4] D.Cox, J.Little, D.O'Shea, "Ideals, Varieties, and Algorithms", Springer-Verlag, 1992.
- [5] S.D.Galbraith, S.Paulus, and N.P.Smart "Arithmetic on Superelliptic Curves", J. Cryptology (1999) 12, 193-196.
- [6] M. Gonda, K. Matsuo, K. Aoki, J. Chao, and S. Tsujii, "Improvements of addition algorithm on genus 3 hyperelliptic curves and their implementations", Proc. of SCIS2004, January 2004.
- [7] R.Harasawa, J.Suzuki, "A Fast Jacobian Group Arithmetic Scheme for Algebraic Curve Cryptography", IEICE TRANS. FOUND., Vol.E84-A No.1, pp.130-139, 2001
- [8] R. Harley, <http://crystal.inria.fr/harley/hyper/adding.text>
- [9] R.Hartshorne, "Algebraic Geometry", Springer-Verlag, 1977.
- [10] J. Kuroki, M. Gonda, K. Matsuo, J. Chao, and S. Tsujii, "Fast genus three hyperelliptic curve cryptosystems", Proc. of SCIS2002, pp. 503–507, January 2002.
- [11] T. Lange, "Weighted Coordinates on Genus 2 Hyperelliptic Curves", IACR ePrint Archive, 2002/153, 2002.
- [12] R. Matsumoto, "The Cab Curve — a generalization of the Weierstrass form to arbitrary plane curves", <http://www.rmatsumoto.org/cab.html>
- [13] K. Matsuo, J. Chao, and S. Tsujii, "Fast genus two hyperelliptic curve cryptosystems", Technical Report ISEC2001-31, IEICE Japan, July 2001.
- [14] S. Miura, "Linear Codes on Affine Algebraic Curves", Trans. of IEICE, vol. J81-A, No. 10, 1398-1421, Oct. 1998.
- [15] J.Pelzl, T.Wollinger, J.Guajardo and C.Paar, "Hyperelliptic Curve Cryptosystems: Closing the Performance Gap to Elliptic Curves," CHES 2003, LNCS 2779, pp.351-365, 2003.



Seigo Arita was born in 1963. He has been interested in prime numbers, algebraic curves and cryptography. He is with Institute of Information Security, Kanagawa, Japan. He is a member of IEICE and JMS.