# Inversion-Free Arithmetic on Genus 3 Hyperelliptic Curves

Xinxin Fan[1] , Yumin Wang[1]

National Key Lab of Integrated Service Networks,
Xidian University,
Xi'an, China,
xxfan@mail.xidian.edu.cn
ymwang@xidian.edu.cn

**Abstract.** Hyperelliptic curve cryptosystem (HECC) is becoming more and more promising for network security applications because of the common effort of several academic and industrial organizations. With short operand size compared to other public key cryptosystems, HECC has showed excellent performance in embedded processors. Recently years, many effort has been made to investigate all kinds of explicit formulae for speeding up group operation of HECC. In this paper, explicit formulae without using inversion for genus 3 HECC are given. We introduce a further coordinate to collect the common denominator of the usual 6 coordinates. The proposed formulae can be used in smart card where inversion is much more expensive than multiplication.

**Keywords:** Genus 3 Hyperelliptic Curve Cryptosystem, Explicit Formulae, Inversion-Free

## 1 Introduction

In 1989, Neal Koblitz proposed using the Jacobian of a hyperelliptic curve defined over a finite field to implement discrete logarithm cryptographic protocols. Due to the work of Cantor [1] (for odd characteristic only) and Koblitz [2], it is possible to perform efficient operation in the ideal class group of a hyperelliptic curve and use HECC in practice.

Recently years, using explicit formulae instead of Cantor algorithm has reduced sharply the complexity of arithmetic in the ideal class group of hyperelliptic curves and obtained fast implementation in software and hardware platform. In the remainder of the paper $I$ denotes a field inversion, $M$ a field multiplication, and $S$ a field squaring.

For genus 2 hyperelliptic curve, Tanja Lange [3] gave explicit formulae which need $1I + 22M + 3S$ for a group addition and $1I + 22M + 5S$ for a group doubling in affine coordinate system. In [4,5,6], the authors introduced a further coordinate called $Z$ to represent the elements of the divisor class group and

[1] Supported by the National NKBRSF '973' Program of China (Grant No.G1999035803)

obtained explicit formulae in projective coordinate system. In [5], $47M + 4S$ are required for a group addition and $40M + 6S$ for a group doubling. In [7], Lange noticed the difference between the denominator of the $V_i$'s and the $U_i$'s and introduced weighted coordinates for genus 2 curves. In addition, the paper found optimal matches to use mixed coordinates and obtained more efficient inversion-free explicit formulae.

For genus 3 case, Kuroki et al. [8] proposed an extension of Harley algorithm over odd characteristic fields and Pelzl et al. [9] gave its improvement and generalization for arbitrary characteristic fields. In [10], the authors used Toom's multiplication to improve Harley's algorithm and showed implementation results on the 64-bit CPU Alpha EV68 1.25GHz. Without distinguishing multiplication and squaring, their proposed algorithm will cost $I+70M$ and $I+71M$ for a group addition and doubling respectively. Affine coordinates are the only coordinate system currently available for genus 3 curves.

In this paper, based on the very recent explicit formulae for addition on genus 3 hyperelliptic curves proposed by Gonda et al. [10], we give inversion-free explicit formulae in projective coordinate system. We will take the same method as in [4,5,6]. In addition, we consider also a group addition with mixed coordinates: one of the input divisor class is represented with affine coordinates, the other projective coordinates and the output divisor class is in projective representation. This kind of mixed addition can also be used efficiently in scalar multiplication.

This paper is organized as follows. Section 2 gives a brief overview of the mathematical background related to genus 3 hyperelliptic curves. In section 3,4 and 5, we investigate inversion-free explicit formulae for group addition, mixed addition and doubling for genus 3 HECC respectively. Finally, Section 6 concludes the whole paper.

## 2 Mathematical Background

In this section we present a brief introduction to some of the theory of genus 3 hyperelliptic curves over odd characteristic fields. For more details the reader is referred to [1], [11] and [12].

Let $n$ be a positive integer, $p \neq 2, 7$ be a prime number, and $q = p^n$. A genus 3 hyperelliptic curve $C$ over $K = GF(q)$ is defined as follows: $C : Y^2 = F(X)$, where $F(X) = X^7 + f_5X^5 + f_4X^4 + f_3X^3 + f_2X^2 + f_1X + f_0 \in GF(p)[X]$ with disc$(F) \neq 0$.

The divisor class group $J_C(GF(q))$ of $C$ forms a finite Abelian group and therefore we can construct cryptosystems based on discrete logarithm problems on the Jacobian of C. Due to Mumford [13], an element of Jacobian, which is called a reduced divisor, has a nice cannonical representation by means of two polynomials $u$ and $v$ defined over $GF(q)$. We will use the notation $[u, v]$ for the divisor represented by $u$ and $v$. For genus 3 curves, we have commonly $[u, v] = [x^3+u_2x^2+u_1x+u_0, v_2x^2+v_1x+v_0]$ with $u|f-v^2$. When using explicit formulae proposed by Gonda et al. [10] to add two reduced divisor classes or double one reduced class, we will obtain another reduced divisor class $[u^{'}, v^{'}] = [x^3 + u^{'}_2 x^2 + u^{'}_1 x + u^{'}_0, v^{'}_2 x^2 + v^{'}_1 x + v^{'}_0]$. For each addition and doubling above, we need one inversion. In order to avoid inversion, we introduce a further coordinate $Z$ to collect the common denominator of the usual 6 coordinates and let the septuple $[U_2, U_1, U_0, V_2, V_1, V_0, Z]$ stand for $[x^3+(U_2/Z)x^2+(U_1/Z)x+(U_0/Z), (V_2/Z)x^2+(V_1/Z)x+(V_0/Z)]$. After finishing scalar multiplication, we will require $I+6M$ to transform the output reduced divisor class from projective coordinates to affine coordinates. In this paper, we study only the most frequent case in detail. Other special cases occur with very low probability and so we will deal with any special case with a less efficient routine (such as Cantor algorithm).

## 3 Inversion-Free Addition Formula

In this section, we give explicit formula for adding two reduced divisor classed in projective coordinate system. When inversions are much slower than multiplications (such as in smart card), we will consider to use inversion-free addition formula. Our formula can also be used for affine inputs if we regard $[u_1, v_1]$ as the septuple $[u_{12}, u_{11}, u_{10}, v_{12}, v_{11}, v_{10}, 1]$. Table 1 lists the number of field operations required to finish each step. If one of the input divisor class is represented in affine coordinates and the other in projective coordinates, we will discuss this case at length in the next section.

## 4 Inversion-Free Mixed Addition Formula

In this section, we show inversion-free mixed addition formula which takes a reduced affine divisor class and a reduced projective divisor class as the input and a reduced projective divisor class as the output. This kind of formula has been widely used in many scalar multiplication algorithms such as (signed) double-and-add, NAF and so on. When using these scalar multiplication algorithms, one of the input is the base divisor class in affine representation and the intermediate result in projective representation. We can see clearly that this kind of addition formula can do better than the previous algorithm. Table 2 lists the number of field operations required to perform the respective steps.

**[Remark]:**Using the formula in Table 2, one saves $31M + 1S$ more than the general inversion-free addition formula. Therefore, when we compute scalar multiplication, it is more efficient to use mixed addition formula.

## 5 Inversion-Free Doubling Formula

In this section, we investigate inversion-free doubling formula. For doubling algorithm the input is almost always in projective representation. Table 3 lists the number of field operations required to complete the respective steps.

**Table 1.** **Explicit Formula For Addition On Genus 3 HEC (Most Frequent Case)**

| | |
|---|---|
| In. | Genus 3 HEC $C: Y^2 = F(X), F = X^7 + f_5X^5 + f_4X^4 + f_3X^3 + f_2X^2 + f_1X + f_0$<br>Reduced Divisors $D_1 = [U_{12}, U_{11}, U_{10}, V_{12}, V_{11}, V_{10}, Z_1]$, $D_2 = [U_{22}, U_{21}, U_{20}, V_{22}, V_{21}, V_{20}, Z_2]$ |
| Out. | $D^{'} = [U^{'}_2, U^{'}_1, U^{'}_0, V^{'}_2, V^{'}_1, V^{'}_0, Z^{'}] = D_1 + D_2(Affine + Affine)$ |

| Step | Expression | Operations |
|---|---|---|
| 1 | Precomputation:<br>$Z = Z_1Z_2, \tilde{Z} = Z^2;$<br>$\tilde{U}_{12} = Z_2U_{12}, \tilde{U}_{11} = Z_2U_{11}, \tilde{U}_{10} = Z_2U_{10}, \tilde{V}_{12} = Z_2V_{12}, \tilde{V}_{11} = Z_2V_{11}, \tilde{V}_{10} = Z_2V_{10};$<br>$\tilde{U}_{22} = Z_1U_{22}, \tilde{U}_{21} = Z_1U_{21}, \tilde{U}_{20} = Z_1U_{20}, \tilde{V}_{22} = Z_1V_{22}, \tilde{V}_{21} = Z_1V_{21}, \tilde{V}_{20} = Z_1V_{20};$ | $13M + S$ |
| 2 | Compute the resultant $r$ of $U_1$ and $U_2$:<br>$t_1 = U_{11}U_{20} - U_{10}U_{21}, t_2 = U_{12}U_{20} - U_{10}U_{22}, t_3 = \tilde{U}_{20} - \tilde{U}_{10}, t_4 = \tilde{U}_{21} - \tilde{U}_{11};$<br>$t_5 = \tilde{U}_{22} - \tilde{U}_{12}, t_6 = t_4^2, t_7 = t_3t_4, t_8 = U_{12}U_{21} - U_{11}U_{22} + t_3, t_9 = t_3^2 - t_1t_5;$<br>$t_{10} = t_2t_5 - t_7, r = [t_8t_9 + t_2(t_{10} - t_7) + t_1t_6]\tilde{Z};$ | $13M + 2S$ |
| 3 | If $r = 0$ then call the Cantor algorithm | |
| 4 | Compute the pseudo-inverse $I = i_2X^2 + i_1X + i_0 \equiv r/U_1 mod U_2$:<br>$i_2 = t_5t_8 - t_6, i_1 = \tilde{U}_{22}i_2 - Zt_{10}, i_0 = \tilde{U}_{21}i_2 - (\tilde{U}_{22}t_{10} + Zt_9);$ | $6M$ |
| 5 | Compute $S^{'} = s^{'}_2X^2 + s^{'}_1X + s^{'}_0 = rS \equiv (V_2 - V_1)I mod U_2$:<br>$t_1 = \tilde{V}_{10} - \tilde{V}_{20}, t_2 = \tilde{V}_{11} - \tilde{V}_{21}, t_3 = \tilde{V}_{12} - \tilde{V}_{22}, t_4 = t_2i_1, t_5 = t_1i_0, t_6 = t_3i_2;$<br>$t_7 = \tilde{U}_{22}t_6, t_8 = t_4 + t_7 + Zt_6 - (t_2 + t_3)(i_1 + Zi_2), t_9 = \tilde{U}_{20} + \tilde{U}_{22};$<br>$t_{10} = (t_9 + \tilde{U}_{21})(t_8 - Zt_6), t_9 = (t_9 - \tilde{U}_{21})(t_8 + Zt_6), s^{'}_0 = -(\tilde{U}_{20}t_8 + Zt_5);$<br>$s^{'}_1 = Z[t_4 + t_5 - t_7 + (t_1 + t_2)(i_0 + i_1)] + (t_9 - t_{10})/2;$<br>$s^{'}_2 = Z[Zt_6 - t_4 - (t_1 + t_3)(i_0 + Zi_2)] - s^{'}_0 - (t_9 + t_{10})/2;$ | $14M$ |
| 6 | If $s^{'}_2 = 0$ then call the Cantor algorithm | |
| 7 | Monic $S = X^2 + (s^{'}_1/s^{'}_2)X + s^{'}_0/s^{'}_2$ | |
| 8 | Precomputation:<br>$\tilde{s}^{'}_0 = s^{'}_0Z, \tilde{s}^{'}_1 = s^{'}_1Z, \tilde{s}^{'}_2 = s^{'}_2Z, \tilde{\tilde{s}}^{'}_2 = \tilde{s}^{'2}_2, s^{"}_2 = \tilde{s}^{'}_2\tilde{Z}, R = rZ;$<br>$A = \tilde{\tilde{s}}^{'}_2\tilde{Z}, B = As^{"}_2, C = A^2, D = RB, E = AD^2, F = AE;$ | $10M + 3S$ |
| 9 | Compute $Z = X^5 + z_4X^4 + z_3X^3 + z_2X^2 + z_1X + z_0 = SU_1$:<br>$z_0 = s^{'}_0\tilde{U}_{10}, z_1 = (s^{'}_0 + s^{'}_1)(\tilde{U}_{10} + \tilde{U}_{11}) - s^{'}_1\tilde{U}_{11} - s^{'}_0\tilde{U}_{10};$<br>$z_2 = (s^{'}_0 + s^{'}_2)(\tilde{U}_{10} + \tilde{U}_{12}) - s^{'}_2\tilde{U}_{12} - s^{'}_0\tilde{U}_{10} - s^{'}_1\tilde{U}_{11};$<br>$z_3 = \tilde{s}^{'}_0 + (s^{'}_1 + s^{'}_2)(\tilde{U}_{12} + \tilde{U}_{11}) - s^{'}_1\tilde{U}_{11} - s^{'}_2\tilde{U}_{12}, z_4 = \tilde{s}^{'}_1 + s^{'}_2\tilde{U}_{12};$ | $8M$ |
| 10 | Compute $U_t = X^4 + u_{t3}X^3 + u_{t2}X^2 + u_{t1}X + u_{t0}$:<br>$u_{t3} = s^{"}_2(z_4 + \tilde{s}^{'}_1 - \tilde{U}_{22}\tilde{s}^{'}_2), t_5 = \tilde{s}^{'}_1z_4 - (\tilde{U}_{22}s^{'}_2)u_{t3};$<br>$u_{t2} = \tilde{Z}[\tilde{s}^{'}_2(z_3 + \tilde{s}^{'}_0 - \tilde{U}_{21}Z) + t_5], t_1 = s^{'}_0z_3;$<br>$t_2 = (\tilde{U}_{22} + \tilde{U}_{21})(\tilde{s}^{'}_2u_{t3} + u_{t2}), t_3 = \tilde{U}_{21}u_{t2}, t_4 = \tilde{Z}t_1 - t_3;$<br>$u_{t1} = \tilde{Z}[\tilde{s}^{'}_2z_2 + (\tilde{s}^{'}_0 + \tilde{s}^{'}_1)(z_3 + z_4) + R(2\tilde{V}_{12}s^{'}_2 - R) - t_5] - Z(t_2 + t_4 + \tilde{\tilde{s}}^{'}_2\tilde{U}_{20});$<br>$u_{t0} = \tilde{Z}(\tilde{s}^{'}_2z_1 + \tilde{s}^{'}_1z_2) + Z\{t_4 + R[2(\tilde{V}_{11}\tilde{s}^{'}_2 + \tilde{V}_{12}\tilde{s}^{'}_1) + R\tilde{U}_{12}]-$<br>$(\tilde{s}^{'}_2u_{t3})\tilde{U}_{20}\} - \tilde{U}_{22}u_{t1};$ | $29M$ |
| 11 | Compute $V_t = v_{t3}X^3 + v_{t2}X^2 + v_{t1}X + v_{t0}$:<br>$t_1 = u_{t3} - s^{"}_2z_4, v_{t0} = t_1u_{t0} + B(z_0 + \tilde{V}_{10}r);$<br>$v_{t1} = t_1u_{t1} + B(z_1 + \tilde{V}_{11}r) - Au_{t0}, v_{t2} = t_1u_{t2} + B(z_2 + \tilde{V}_{12}r) - Au_{t1};$<br>$v_{t3} = t_1u_{t3} + Bz_3 - Au_{t2};$ | $15M$ |
| 12 | Compute $U_3 = X^3 + u_{32}X^2 + u_{31}X + u_{30}$:<br>$t_1 = 2v_{t3}, u_{32} = -(Eu_{t3} + Cv_{t3}^2), u_{31} = f_5F - Eu_{t2} - Ct_1v_{t2} - (Au_{32})u_{t3};$<br>$u_{30} = f_4F - Eu_{t1} - C(t_1v_{t1} + v_{t2}^2) - (Au_{32})u_{t2} - u_{31}u_{t3};$ | $14M + 2S$ |
| 13 | Adjust:<br>$Z^{'} = FD, U^{'}_2 = u_{32}D, U^{'}_1 = u_{31}D, U^{'}_0 = u_{30}D;$ | $4M$ |
| 14 | Compute $V_3 = v_{32}X^2 + v_{31}X + v_{30}$:<br>$V^{'}_2 = Fv_{t2} - u_{32}v_{t3}, V^{'}_1 = Fv_{t1} - u_{31}v_{t3}, V^{'}_0 = Fv_{t0} - u_{30}v_{t3};$ | $6M$ |
| Sum | | $132M + 8S$ |

**Table 2.** **Explicit Formula For Mixed Addition On Genus 3 HEC (Most Frequent Case)**

| | |
|---|---|
| In. | Genus 3 HEC $C : Y^2 = F(X), F = X^7 + f_5X^5 + f_4X^4 + f_3X^3 + f_2X^2 + f_1X + f_0$<br>Reduced Divisors $D_1 = [U_{12}, U_{11}, U_{10}, V_{12}, V_{11}, V_{10}, Z_1], \quad D_2 = [U_{22}, U_{21}, U_{20}, V_{22}, V_{21}, V_{20}, 1]$ |
| Out. | $D^{'} = [U_2^{'}, U_1^{'}, U_0^{'}, V_2^{'}, V_1^{'}, V_0^{'}, Z^{'}] = D_1 + D_2(Projective + Affine)$ |

| Step | Expression | Operations |
|---|---|---|
| 1 | Precomputation:<br>$\tilde{U}_{22} = Z_1U_{22}, \tilde{U}_{21} = Z_1U_{21}, \tilde{U}_{20} = Z_1U_{20}, \tilde{V}_{22} = Z_1V_{22}, \tilde{V}_{21} = Z_1V_{21}, \tilde{V}_{20} = Z_1V_{20};$ | $6M$ |
| 2 | Compute the resultant $r$ of $U_1$ and $U_2$:<br>$t_1 = U_{11}U_{20} - U_{10}U_{21}, t_2 = U_{12}U_{20} - U_{10}U_{22}, t_3 = \tilde{U}_{20} - U_{10}, t_4 = \tilde{U}_{21} - U_{11};$<br>$t_5 = \tilde{U}_{22} - U_{12}, t_6 = t_4^2, t_7 = t_3t_4, t_8 = U_{12}U_{21} - U_{11}U_{22} + t_3, t_9 = t_3^2 - t_1t_5;$<br>$t_{10} = t_2t_5 - t_7, r = t_8t_9 + t_2(t_{10} - t_7) + t_1t_6;$ | $12M + 2S$ |
| 3 | If $r = 0$ then call the Cantor algorithm | |
| 4 | Compute the pseudo-inverse $I = i_2X^2 + i_1X + i_0 \equiv r/U_1 mod U_2$:<br>$i_2 = t_5t_8 - t_6, i_1 = U_{22}i_2 - t_{10}, i_0 = U_{21}i_2 - U_{22}t_{10} - t_9;$ | $4M$ |
| 5 | Compute $S^{'} = s_2^{'}X^2 + s_1^{'}X + s_0^{'} = rS \equiv (V_2 - V_1)I mod U_2$:<br>$t_1 = V_{10} - \tilde{V}_{20}, t_2 = V_{11} - \tilde{V}_{21}, t_3 = V_{12} - \tilde{V}_{22}, t_4 = t_2i_1, t_5 = t_1i_0, t_6 = t_3i_2;$<br>$t_7 = U_{22}t_6, t_8 = t_4 + t_7 + t_6 - (t_2 + t_3)(i_1 + i_2), t_9 = U_{20} + U_{22};$<br>$t_{10} = (t_9 + U_{21})(t_8 - t_6), t_9 = (t_9 - U_{21})(t_8 + t_6), s_0^{'} = -(U_{20}t_8 + t_5);$<br>$s_1^{'} = t_4 + t_5 + (t_9 - t_{10})/2 - t_7 - (t_1 + t_2)(i_0 + i_1);$<br>$s_2^{'} = t_6 - s_0^{'} - (t_9 + t_{10})/2 - t_4 - (t_1 + t_3)(i_0 + i_2);$ | $10M$ |
| 6 | If $s_2^{'} = 0$ then call the Cantor algorithm | |
| 7 | Monic $S = X^2 + (s_1^{'}/s_2^{'})X + s_0^{'}/s_2^{'}$ | |
| 8 | Precomputation:<br>$\tilde{s}_0^{'} = s_0^{'}Z_1, \tilde{s}_1^{'} = s_1^{'}Z_1, \tilde{s}_2^{'} = s_2^{'}Z_1, s_2^{''} = s_2^{'2}, R = rZ_1;$<br>$A = s_2^{''}Z_1, B = As_2^{'}, C = A^2, D = RB, E = AD^2, F = AE;$ | $9M + 3S$ |
| 9 | Compute $Z = X^5 + z_4X^4 + z_3X^3 + z_2X^2 + z_1X + z_0 = SU_1$:<br>$z_0 = s_0^{'}U_{10}, z_1 = (s_0^{'} + s_1^{'})(U_{10} + U_{11}) - s_1^{'}U_{11} - s_0^{'}U_{10};$<br>$z_2 = (s_0^{'} + s_2^{'})(U_{10} + U_{12}) - s_2^{'}U_{12} - s_0^{'}U_{10} - s_1^{'}U_{11};$<br>$z_3 = \tilde{s}_0^{'} + (s_1^{'} + s_2^{'})(U_{12} + U_{11}) - s_1^{'}U_{11} - s_2^{'}U_{12}, z_4 = \tilde{s}_1^{'} + s_2^{'}U_{12};$ | $8M$ |
| 10 | Compute $U_t = X^4 + u_{t3}X^3 + u_{t2}X^2 + u_{t1}X + u_{t0}$:<br>$u_{t3} = s_2^{'}z_4 + s_2^{'}\tilde{s}_1^{'} - \tilde{U}_{22}s_2^{''}, t_5 = s_1^{'}z_4 - (s_2^{'}u_{t3})U_{22};$<br>$u_{t2} = s_2^{'}(z_3 + \tilde{s}_0^{'} - U_{21}\tilde{s}_2^{'}) + t_5, t_1 = s_0^{'}z_3;$<br>$t_2 = (U_{22} + U_{21})(s_2^{'}u_{t3} + u_{t2}), t_3 = U_{21}u_{t2}, t_4 = t_1 - t_3;$<br>$u_{t1} = s_2^{'}z_2 + (s_0^{'} + s_1^{'})(z_3 + z_4) + r(2V_{12}s_2^{'} - R) - (t_5 + t_2 + t_4 + s_2^{''}\tilde{U}_{20});$<br>$u_{t0} = s_2^{'}z_1 + s_1^{'}z_2 + t_4 + r[2(V_{11}s_2^{'} + V_{12}s_1^{'}) + rU_{12}] - (s_2^{'}u_{t3})U_{20} - U_{22}u_{t1};$ | $24M$ |
| 11 | Compute $V_t = v_{t3}X^3 + v_{t2}X^2 + v_{t1}X + v_{t0}$:<br>$t_1 = u_{t3} - s_2^{'}z_4, v_{t0} = t_1u_{t0} + B(z_0 + V_{10}r);$<br>$v_{t1} = t_1u_{t1} + B(z_1 + V_{11}r) - Au_{t0}, v_{t2} = t_1u_{t2} + B(z_2 + V_{12}r) - Au_{t1};$<br>$v_{t3} = t_1u_{t3} + Bz_3 - Au_{t2};$ | $14M$ |
| 12 | Compute $U_3 = X^3 + u_{32}X^2 + u_{31}X + u_{30}$:<br>$t_1 = 2v_{t3}, u_{32} = -(Eu_{t3} + Cv_{t3}^2), u_{31} = f_5F - Eu_{t2} - Ct_1v_{t2} - (Au_{32})u_{t3};$<br>$u_{30} = f_4F - Eu_{t1} - C(t_1v_{t1} + v_{t2}^2) - (Au_{32})u_{t2} - u_{31}u_{t3};$ | $14M + 2S$ |
| 13 | Adjust:<br>$Z^{'} = FD, U_2^{'} = u_{32}D, U_1^{'} = u_{31}D, U_0^{'} = u_{30}D;$ | $4M$ |
| 14 | Compute $V_3 = v_{32}X^2 + v_{31}X + v_{30}$:<br>$V_2^{'} = Fv_{t2} - u_{32}v_{t3}, V_1^{'} = Fv_{t1} - u_{31}v_{t3}, V_0^{'} = Fv_{t0} - u_{30}v_{t3};$ | $6M$ |
| Sum | | $101M + 7S$ |

**Table 3.** **Explicit Formula For Doubling On Genus 3 HEC (Most Frequent Case)**

| | |
|---|---|
| In. | Genus 3 HEC $C$ : $Y^2 = F(X)$, $F = X^7 + f_5X^5 + f_4X^4 + f_3X^3 + f_2X^2 + f_1X + f_0$<br>Reduced Divisors $D_1 = [U_{12}, U_{11}, U_{10}, V_{12}, V_{11}, V_{10}, Z]$ |
| Out. | $D' = [U_2', U_1', U_0', V_2', V_1', V_0', Z'] = 2D_1(Projective)$ |

| Step | Expression | Operations |
|---|---|---|
| 1 | Precomputation:<br>$\tilde{Z} = Z^2, \tilde{\tilde{Z}} = \tilde{Z}^2$;<br>$\tilde{U}_{12} = ZU_{12}, \tilde{U}_{11} = ZU_{11}, \tilde{U}_{10} = ZU_{10}, \tilde{V}_{12} = ZV_{12}, \tilde{V}_{11} = ZV_{11}, \tilde{V}_{10} = ZV_{10}$; | $6M + 2S$ |
| 2 | Compute the resultant $r$ of $U_1$ and $V_1$:<br>$t_1 = U_{11}V_{10} - U_{10}V_{11}, t_2 = U_{12}V_{10} - U_{10}V_{12}, t_3 = V_{11}^2, t_4 = V_{11}V_{10}$;<br>$t_5 = \tilde{V}_{10} + U_{12}V_{11} - U_{11}V_{12}, t_6 = \tilde{V}_{10}V_{10} - V_{12}t_1, t_7 = V_{12}t_2 - Zt_4$;<br>$r = [t_5t_6 + t_2(t_7 - Zt_4) + (Zt_3)t_1]Z\tilde{\tilde{Z}}$; | $17M + 1S$ |
| 3 | If $r = 0$ then call the Cantor algorithm | |
| 4 | Compute the pseudo-inverse $I = i_2X^2 + i_1X + i_0 \equiv r/V_1 mod U_1$:<br>$i_2 = Z(Zt_3 - V_{12}t_5), i_1 = U_{12}i_2 + Zt_7, i_0 = U_{11}i_2 + U_{12}t_7 + Zt_6$; | $7M$ |
| 5 | Compute $Z = z_2X^2 + z_1X + z_0 \equiv (F - V_1^2)/U_1 mod U_1$:<br>$t_3 = U_{12}^2, t_4 = f_4\tilde{Z} - (2\tilde{U}_{10} + V_{12}^2), t_5 = f_5\tilde{Z} + t_3 - 2\tilde{U}_{11}$;<br>$z_2 = \tilde{Z}(t_5 + 2t_3), z_1 = \tilde{U}_{12}(2\tilde{U}_{11} - t_5) + \tilde{Z}t_4$;<br>$z_0 = f_3\tilde{\tilde{Z}} + t_3(t_5 - \tilde{U}_{11}) + \tilde{U}_{12}(2\tilde{U}_{10} - t_4) + \tilde{U}_{11}(\tilde{U}_{11} - f_5\tilde{Z}) - 2\tilde{V}_{12}\tilde{V}_{11}$; | $10M + 2S$ |
| 6 | Compute $S' = s_2'X^2 + s_1'X + s_0' = 2rS \equiv ZI mod U_1$:<br>$t_1 = i_1z_1, t_2 = i_0z_0, t_3 = i_2z_2, t_4 = U_{12}t_3$;<br>$t_5 = Z[(i_2 + i_1)(z_2 + z_1) - (t_1 + t_3)] - t_4, t_6 = U_{10}t_5, t_7 = U_{10} + U_{12}$;<br>$t_8 = t_7 + U_{11}, t_9 = t_7 - U_{11}, t_7 = t_8(Zt_3 + t_5), t_{11} = t_9(t_5 - Zt_3)$;<br>$s_2' = \tilde{Z}[t_1 - t_2 - t_3 + (i_2 + i_0)(z_2 + z_0)] + t_6 - (t_7 + t_{11})/2$;<br>$s_1' = \tilde{Z}[(i_0 + i_1)(z_0 + z_1) - (t_1 + t_2)] + Zt_4 + (t_{11} - t_7)/2, s_0' = \tilde{Z}t_2 - t_6$; | $16M$ |
| 7 | If $s_2' = 0$ then call the Cantor algorithm | |
| 8 | Monic $S = X^2 + (s_1'/s_2')X + s_0'/s_2'$ | |
| 9 | Precomputation:<br>$\tilde{s}_0' = s_0'Z, \tilde{s}_1' = s_1'Z, \tilde{s}_2' = s_2'Z, R = rZ$;<br>$A = \tilde{s}_2'^2, B = A\tilde{s}_2', C = A^2, D = 2RB, E = AD^2, F = AE$; | $8M + 3S$ |
| 10 | Compute $G = X^5 + g_4X^4 + g_3X^3 + g_2X^2 + g_1X + g_0 = SU_1$:<br>$g_0 = s_0'U_{10}, g_1 = (s_0' + s_1')(U_{10} + U_{11}) - s_1'U_{11} - s_0'U_{10}$;<br>$g_2 = (s_0' + s_2')(U_{10} + U_{12}) - s_2'U_{12} - s_0'U_{10} - s_1'U_{11}$;<br>$g_3 = \tilde{s}_0' + (s_1' + s_2')(U_{12} + U_{11}) - s_1'U_{11} - s_2'U_{12}, g_4 = \tilde{s}_1' + s_2'U_{12}$; | $8M$ |
| 11 | Compute $U_t = X^4 + u_{t3}X^3 + u_{t2}X^2 + u_{t1}X + u_{t0}$:<br>$u_{t3} = 2\tilde{s}_1'\tilde{s}_2', u_{t2} = \tilde{s}_1'^2 + 2\tilde{s}_0'\tilde{s}_2', u_{t1} = 2[\tilde{s}_1'\tilde{s}_0' + 2R(s_2'V_{12} - R)]$;<br>$u_{t0} = \tilde{s}_0'^2 + 4r[U_{12}(2R - s_2'V_{12}) + s_1'\tilde{V}_{12} + s_2'\tilde{V}_{11}]$; | $9M + 2S$ |
| 12 | Compute $V_t = v_{t3}X^3 + v_{t2}X^2 + v_{t1}X + v_{t0}$:<br>$t_1 = u_{t3} - \tilde{s}_2'g_4, v_{t0} = t_1u_{t0} + B(g_0 + V_{10}r)$;<br>$v_{t1} = t_1u_{t1} + B(g_1 + V_{11}r) - Au_{t0}, v_{t2} = t_1u_{t2} + B(g_2 + V_{12}r) - Au_{t1}$;<br>$v_{t3} = t_1u_{t3} + Bg_3 - Au_{t2}$; | $15M$ |
| 13 | Compute $U_3 = X^3 + u_{32}X^2 + u_{31}X + u_{30}$:<br>$t_1 = 2v_{t3}, u_{22} = -(Eu_{t3} + Cv_{t3}^2), u_{21} = f_5F - Eu_{t2} - Ct_1v_{t2} - (Au_{32})u_{t3}$;<br>$u_{20} = f_4F - Eu_{t1} - C(t_1v_{t1} + v_{t2}^2) - (Au_{32})u_{t2} - u_{31}u_{t3}$; | $14M + 2S$ |
| 14 | Adjust:<br>$Z' = FD, U_2' = u_{22}D, U_1' = u_{21}D, U_0' = u_{20}D$; | $4M$ |
| 15 | Compute $V_3 = v_{32}X^2 + v_{31}X + v_{30}$:<br>$V_2' = Fv_{t2} - u_{22}v_{t3}, V_1' = Fv_{t1} - u_{21}v_{t3}, V_0' = Fv_{t0} - u_{20}v_{t3}$; | $6M$ |
| Sum | | $120M + 12S$ |

## 6 Conclusion And Outlook

We gave explicit formulae to perform inversion free arithmetic on genus 3 hyperelliptic curve. Our Explicit formulae cost respectively $132M + 8M$, $101M + 7S$ and $120M + 12S$ to perform addition, mixed addition and doubling. The practical performance of our algorithm in embedded system (especially in smart card) needs studying further.

In order to minimize the number of operations, we did not keep the additional coordinate $Z^{'}$ minimal as in [4]. We took respectively $Z^{'} = r^3 s_2^{'13} Z^{32}$, $r^3 s_2^{'13} Z_1^8$ and $8r^3 s_2^{'13} Z^{16}$ for addition, mixed addition and doubling formula. Furthermore, we must adjust the denominator of $U_2^{'}$, $U_1^{'}$ and $U_0^{'}$ to be the same as that of $V_2^{'}$, $V_1^{'}$ and $V_0^{'}$.

How to improve our algorithms and to generalize weighted projective coordinates to genus 3 curves will be our important future works.

## References

1. D.G.Cantor.: Computing In The Jacobian Of A Hyperelliptic Curve. Math. Comp., 48:95-101, 1987.
2. N.Koblitz.: Hyperelliptic Cryptosystems. In Ernest F.Brickell, editor, Journal of Cryptology, pp.139-150, 1989.
3. T.Lange.: Efficient Arithmetic on Genus 2 Hyperelliptic Curve Over Finite Field via Explicit Formulae. Cryptology ePrint Archieve, Report 2002/121, http://eprint.iacr.org/, 2002
4. T.Lange.: Inversion-Free Arithmetic on Genus 2 Hyperelliptic Curves. Cryptology ePrint Archieve, Report 2002/147, http://eprint.iacr.org/, 2002
5. Y.Miyamato, H.Doi, K.Matsuo, J.Chao and S.Tsujii.: A Fast Addition Algorithm Of Genus Two Hyperelliptic Curve. In Proc. of SCIS 2002, IEICE Japan, pp.497-502, 2002. in Japanese.
6. T.Lange.: Formulae For Arithmetic On Genus 2 Hyperelliptic Curves. http://www.ruhr-uni-bochum.de/itsc/tanja/preprints/expl-sub.pdf., 2003
7. T.Lange.: Weighted Coordinates On Genus 2 Hyperelliptic Curves. Cryptology ePrint Archieve, Report 2002/153, http://eprint.iacr.org/, 2002
8. J.Kuroki, M.Gonda, K.Matsuo, J.Chao and S.Tsujii.: Fast Genus Three Hyperelliptic Curve Cryptosystems. In Proc. of SCIS 2002, IEICE Japan, pp.503-507, 2002
9. J.pelzl, T.Wollinger, J.Guajardo and C.Paar, Hyperelliptic Curve Cryptosystems: Closing The Performance Gap To elliptic Curve (Update), Cryptology ePrint Archieve, Report 2003/026, http://eprint.iacr.org/, 2003
10. M.Gonda, K.Matsuo, K.Aoki, J.Chao and S.Tsujii.: Improvements Of Addition Algorithm On Genus 3 Hyperelliptic Curves And Their Implementations. In Proc. of SCIS 2004, Japan, 2004
11. N.Koblitz.: Hyperelliptic Cryptosystems. In Ernest F.Brickell, editor, Journal of Cryptology, pp.139–150, 1989.
12. A.Menezes, Y.Wu and R.Zuccherato.: An Elementary Introduction to Hyperelliptic Curve. Technical Report CORR 96-19, University of Waterloo, 1996, Canada. Available at http://www.cacr.math.uwaterloo.ca
13. D.Mumford.: Tata Lectures on Theta II. Birkhäuser, 1983

**About the author:**

**Xinxin Fan:** was born in 1980. He received the B.S. degree in applied mathematics in 2002 from Xidian University, China. He is a M.S. candidate in the National Key Lab On Integrated Services Networks,Xidian University. His research interests include elliptic curve cryptography, hyperelliptic curve cryptography and side-channel attack.

**Yumin Wang:** was born in 1936. He received the B.E. degree from Department of Telecommunication Engineering, Xidian University, China in 1959. In 1979-1981, he was a visiting scholar in Department of Electronic Engineering, Hawaii University. Currently he is a professor, a Ph.D. supervisor in Xidian University. He is a fellow member of the Chinese Institute of Communication, a fellow member of the Chinese Institute of Electronics. He serves as a member of the Board of Governors of the Chinese Institute of Cryptography (preparator committee) and also serves on the committee of Information Theory Society for the Chinese Institute of Electronics, and a senior member of IEEE. His research interests are communication, information theory, coding and cryptography.