Classification of genus 2 curves over $\mathbb{F}_{2^n}$ and optimization of their arithmetic

Bertrand Byramjee[1] and Sylvain Duquesne[2]

[1] Oberthur Card Systems,
25, rue Auguste Blanche, 92800 Puteaux, France,
b.byramjee@oberthurcs.com
[2] Université de Bordeaux I, Laboratoire A2X
351 Cours de la libération, 33405 Talence Cedex, France,
duquesne@math.u-bordeaux.fr

Abstract

To obtain efficient cryptosystems based on hyperelliptic curves, we studied genus 2 isomorphism classes of hyperelliptic curves in characteristic 2. We found general and optimal form for these curves, just as the short Weierstrass form for elliptic curves. We studied the security and the arithmetic on their jacobian. We also rewrote and optimized the formulas of Lange in characteristic 2, and we introduced a new system of coordinate. Therefore, we deduced the best form of hyperelliptic curves of genus 2 in characteristic 2 to use in cryptography.

Key words. hyperelliptic curve cryptography, genus 2, characteristic 2, explicit formulas, security, isomorphism classes, standardization of curves.

1 Introduction

There is no sub-exponential time algorithm to solve the discrete problem based on abelian generic group. Elliptic curves provide the simplest example with no better algorithm than for generic group. In 1985, Elliptic curves cryptosystems were introduced independently by Miller [13] and Koblitz [6]. In 1989, Koblitz [7] suggested using the jacobian of hyperelliptic curves as a source of finite abelian groups. The main advantage is to use smaller ground field for the same level of security. For example, a hyperelliptic curve of genus 2 over $\mathbb{F}_{2^{80}}$ can be used in order to have the same level of security as an elliptic curve defined over $\mathbb{F}_{2^{160}}$.

This paper deals with hyperelliptic curves of genus 2 in characteristic 2. It is organized as follows. In section 2, we recall the basic notions of hyperelliptic curves. We refer the reader to [8] for further details and in section 3 we proceed as in [2] and [16] to classify hyperelliptic curves. In the case of elliptic curves ($g = 1$), one can prove that every non supersingular curve can be transformed into a curve of the type:

$$y^2 + xy = x^3 + a_2x^2 + a_6.$$

At this point, there is no analogous in higher genus. Such a representation is very important to define a standard for hyperelliptic curves. Some work has already been done in this field, at least in genus 2 in [2]. Nevertheless we can improve it. We suggest two types of curves suitable for cryptography which are general and optimal in a sense that we will precise later.

In section 4, we analyze the security of the different classes of curves defined in the previous section. In section 5, we rewrite and optimize characteristic 2 formulas of Lange, but we count multiplications of all the coefficients. Moreover we suggest a new system of coordinates which allows faster scalar multiplications on jacobians. All these formulas are given in appendix. Thanks to the results of the last two sections, we suggest a form for equations for hyperelliptic curves of genus 2 in characteristic 2 for future standards in cryptography.

2 Background on Hyperelliptic curves

Let $\overline{\mathbb{F}}_{2^n}$ be an algebraic closure of the field $\mathbb{F}_{2^n}$. A hyperelliptic curve C of genus $g \geq 1$ on $\mathbb{F}_{2^n}$ is given by the general equation :

$$C : y^2 + h(x)y = f(x) \tag{G}$$

where $h \in \mathbb{F}_{2^n}[X]$, is a polynomial of degree at most g, $f \in \mathbb{F}_{2^n}[X]$ is a monic polynomial of degree $2g + 1$ and there is no singular points $(x, y) \in \overline{\mathbb{F}}_{2^n} \times \overline{\mathbb{F}}_{2^n}$. These are the solutions satisfying simultaneously equation (G) and the partial derivative equations $h(x) = 0$ and $h'(x)y + f'(x) = 0$.

Now, we concentrate in the genus 2 case. Let us define some objects on these curves.

A divisor D is a formal sum of points on the hyperelliptic curve C. The jacobian J is the group of degree 0 divisors modulo principal divisors. In practice, we use the Mumford representation : each divisor is represented by a pair of polynomials $[u, v]$ such that u is a monic polynomial of degree 2, deg $v <$ deg u and $u|f - hv - v^2$ (these types of divisors are called reduced).

Cantor described a general algorithm (working in every genera) to add divisors on J, see [1] for more definitions on hyperelliptic curves and details on this algorithm. Nevertheless, his algorithm is too slow, mainly because using gcd algorithms, and uses up too much memory for restricted environments like smart cards.

To improve it in the genus 2 case, Lange following Harley [5], suggests several explicit formulas in affine, projective and weighted projective, in [9], [10] & [11]. Nevertheless she doesn't count multiplications by the coefficients of h, as with the Koblitz curves. Therefore her formulas are not general. That's why we suggest here to rewrite her formulas in the general case and in the different types we define in section 2. In so doing, we optimize these formulas. The best optimizations we obtained, are in the doubling case which is the most important in scalar multiplication.

3 Classification of genus 2 hyperelliptic curves over $\mathbb{F}_{2^n}$

For the genus 2 case, we use the following equation

$$y^2 + (h_2x^2 + h_1x + h_0)y = x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0.$$

We divide the hyperelliptic curves into three types depending on the leading coefficient of h, following the notation of [2]:

- type I: $h_2 \neq 0$.
- type II: $h_2 = 0,\ h_1 \neq 0$.
- type III: $h_2 = h_1 = 0,\ h_0 \neq 0$.

Moreover, Choie and Yun prove in [2] that type I has asymptotically between $2q^3$ and $4q^3$ isomorphism classes ($q = 2^n$), type II about $2q^2$ and type III between $2q$ and $32q$. Nevertheless, from these 3 types, only 2 are interesting for a cryptosystem based on the Discrete Logarithm problem, as Galbraith proves in [4] the following result.

Proposition 1. *A characteristic 2 hyperelliptic curve is of type III if and only if it is supersingular.*

Let us first give results concerning the resolution of some simple equation in $\mathbb{F}_{2^n}$.

Proposition 2. *Let $a, b \in \mathbb{F}_{2^n}$,*

1. *The equation $x^{2^k} = b$ has always a solution in $\mathbb{F}_{2^n}$ for $k \geq 1$.*
2. *The equation $x^3 = b$ has always a solution in $\mathbb{F}_{2^n}$ if n is odd.*
3. *For $a \neq 0$, $x^2 + ax + b = 0$ has a solution in $\mathbb{F}_{2^n}$ iff $Tr(a^{-2}b) = 0$.*
4. *If $Tr(a^{-2}b) = 1$, the equation $x^2 + ax + b = ta^2$ has a solution in $\mathbb{F}_{2^n}$ where t is an element of trace 1.*

Remark:

- Here the Trace function is defined by $\mathrm{Tr}(x) = \sum_{i=0}^{n-1} x^{2^i}$.
- In 4, if n is odd, t can be chosen equal to 1 and if n is even, t is a power of π in polynomial basis representation (i.e. $\mathbb{F}_{2^n} \simeq \mathbb{F}_2[\pi]$). In all cases, multiplication by t is free.

Sketch of the proof:

1. $x^2 = b$ has always the solution $x = b^{2^{n-1}}$. This proves the first point.
2. $x^3 = b$ has a solution in $\mathbb{F}_{2^n}$ iff $b^{\frac{2^n-1}{d}} = 1$ where $d = gcd(2^n - 1, 3)$. If n is odd, $d = 1$ so $x^3 = b$ has a solution.
3. This is an application of the additive form of Hilbert's "Satz 90".
4. Please note that $\mathrm{Tr}(a^{-2}(b + ta^2)) = 0$. $\square$

We will now write equation for type I and type II in a minimal form, in the sense that if the coefficients of the equation describe the base field, the expected number of curves is obtained (say $2q^3$ for type I and $2q^2$ for type II). In the following, t denotes an element of trace 1 ($t = 1$ if n is odd) as explained in the previous proposition and ε an element of $\mathbb{F}_2$.

Theorem 1. *A characteristic 2 hyperelliptic curve of type I can always be transformed into one of the following equations:*

$$\begin{array}{lcl} \textit{type Ia} & : & y^2 + (x^2 + h_1 x + t h_1^2) y = x^5 + t\varepsilon x^4 + f_1 x + f_0, \\ \textit{type Ib} & : & y^2 + x(x + h_1) y = x^5 + t\varepsilon x^4 + f_1 x + f_0. \end{array}$$

Remark:

- It is possible to define only one type, but we chose to separate the case where the polynomial h is irreducible (type Ia) and the case where it can be factorized (type Ib) because they are mathematically different. For example, the order of the jacobian of a type Ia curve will always be divisible by two, (since there exists a divisor of order 2) whereas it is divisible by 4 (since there exists two divisors of order 2) in type Ib case.
  This kind of observation is of course very important in cryptography and must be taken into account if one wants to construct good curves for future standards.

- In both cases, we obtain in this way at most $2q^3$ isomorphism classes of curves of type I, which was the expected number as proved in [2].

Sketch of the proof: specializing Lockhart's formula (see [12] for details),

$$\begin{cases} x = & h_2^2 x + \lambda \\ y = & h_2^5 y + h_2^4 \alpha x^2 + h_2^2 \beta x + \gamma \end{cases}$$

with

- λ a root of $h_2X^2 + h_1X + h_0$, if $\mathrm{Tr}(h_0h_2h_1^{-2}) = 0$ and we obtain a curve of type Ib.

- λ a root of $h_2X^2 + h_1X + h_0 + th_1^2h_2^{-1}$, if $\mathrm{Tr}(h_0h_2h_1^{-2}) = 1$ and we obtain a curve of type Ia.

- α a root of $X^2 + h_2X + f_4 + \lambda + \varepsilon t h_2^2$ with $\varepsilon = \mathrm{Tr}((f_4 + \lambda)h_2^2)$.

- $\beta = (f_3 + h_1\alpha)h_2^{-1}$.

- $\gamma = (\beta^2 + h_1\beta + \alpha(h_2\lambda^2 + h_1\lambda + h_0) + f_3\lambda + f_2)h_2^{-1}$. $\square$

Theorem 2. *If n is odd, a hyperelliptic curve of type II defined over $\mathbb{F}_{2^n}$ can be transformed into the following equation :*

$$y^2 + xy = x^5 + f_3x^3 + \varepsilon x^2 + f_0.$$

Sketch of the proof: with Lockhart's formula:

$$\begin{cases} x = & \mu^2 x + \lambda \\ y = & \mu^5 y + \mu^4 \alpha x^2 + \mu^2 \beta x + \gamma \end{cases}$$

with

- μ such as $\mu^3 = h_1$,
- $\lambda = h_0 h_1^{-1}$,
- $\alpha = \sqrt{\lambda + f_4}$,
- β is a root of $X^2 + h_1 X + f_2 + \varepsilon h_1^2$ with $\varepsilon =$Tr$(f_2 h_1^{-2})$,
- $\gamma = \big((h_0 + h_1\lambda)\beta + \lambda^2 f_3 + \lambda^4 + f_1\big)\, h_1^{-1}$. $\square$

Theorem 2'. *If n is even, a hyperelliptic curve of type II defined over $\mathbb{F}_{2^n}$ can be transformed into the following equation :*

$$y^2 + h_1 xy = x^5 + \varepsilon' x^3 + t\varepsilon h_1^2 x^2 + f_0.$$

Remark:

- To prove theorem 2', one just have to choose μ so that $\mu^4 = f_3 + h_1\alpha$.
- In theorem 2, we could have erased f_3 instead of h_0, choosing λ before α and so would have had the following form:

  $$y^2 + (x + h_0)y = x^5 + \varepsilon x^2 + f_0.$$

  This form can be useful if someone wants to implement a general form of a hyperelliptic curve as there is no f_3 term in type I or type II. Nevertheless we didn't choose this form as we lose performance by keeping h_0 in the explicit formulas.
- To avoid the Weil-descent attack, n must be chosen prime, which means that only the theorem 2 is of interest for cryptographic purposes.
- If n is odd (resp. even), we obtain in this way at the most $2q^2$ (resp. $4q^2$) isomorphism classes of curves of type II, which was the expected number as proved in [2].

If one wants to use pairings, we provide the following result for the last type of hyperelliptic curves.

Theorem 3. *A characteristic 2 hyperelliptic curve of type III can be transformed into the following equation :*

$$y^2 + y = x^5 + f_3 x^3 + f_1 x + t\varepsilon.$$

Sketch of the proof: with Lockhart's formula:

$$\left\{ \begin{array}{rl} x = & \mu^2 x \\ y = & \mu^5 y + \mu^4 \alpha x^2 + \mu^2 \beta x + \gamma \end{array} \right.$$

with

- μ such as $\mu^5 = h_0$,
- $\alpha = \sqrt{f_4}$,
- $\beta = \sqrt{f_2 + f_4}$,
- γ is a root of $X^2 + h_0 X + f_0 + \varepsilon h_0^2$ with $\varepsilon =$Tr$(f_0 h_0^{-2})$. $\square$

Remark:
This is not the optimal form as there are between $2q$ and $32q$ curves of type III. Nevertheless, we believe that the correct choice is to take $f_1 = 0$, but we can't prove it in a general way.

4 Analysis of the security of different types of curves

In the previous section, we have classified the curves of genus 2 define over $\mathbb{F}_{2^n}$. In order to use these curves in cryptography, it is very interesting to check the security of each type of curve and to compare them. For example, in proposition 1, we have already seen that all curves of type III are supersingular, which means that they are weak for cryptographic use.

To compare the behavior of different curves, we computed the cardinality of at least 10 000 curves of each type and each value of ε. We use the implementation of Kedlaya's algorithm to compute the cardinality of the jacobian of a curve of genus 2 [15]. We thank F. Vercauteren for allowing us to use his implementation and for answering kindly all our questions.

We have chosen $\mathbb{F}_{2^{89}}$ as ground field so that all the curves are resistant to Weil descent attacks, see Rück in [14] for details.

We call *good curves* those suitable for cryptography, i.e. where there is a divisor of prime order greater than 2^{160} and *nice curves* those with minimal cofactor. In characteristic 2, as in the case of elliptic curves, the cardinality cannot be prime, but we want the cofactor to be minimal (we denote it by f). For example a nice curve with cofactor 2 means that the cardinality of the jacobian is two times a prime.

For each type of curve, we computed the rate of good curves and the rate of nice curves. Moreover, proposition 1 states that curves of type III are the only supersingular curves. However, this didn't prove that curves of type I or II are resistant to Frey-Rück attack [3] (using transfer via the Tate-Lichtenbaum pairing) but it seems to be true in practice. In fact all the curves we tested are resistant.

	good curves	nice curves	minimal f	curves tested
Type Ia, $\varepsilon = 0$	10.4 %	0.56 %	2	10 000
$\varepsilon = 1$	10 %	0.53 %	2	11 446
Type Ib, $\varepsilon = 0$	8.9 %	0.33 %	4	10 000
$\varepsilon = 1$	9.6 %	0.6 %	4	11 445
Type II, $\varepsilon = 0$	9.6 %	0.6 %	4	20 917
$\varepsilon = 1$	10.9 %	1.23 %	2	16 724

We note with these computations, that there are some differences between different types of curves. We already stated that the order of the jacobian is always divisible by 2 for type Ia and by 4 for type Ib, therefore one could hope to find more good curves of type Ia than Ib. This is in fact the case. We can conclude that if one wants to use curves of type I, it is better to choose type Ia because there are more good curves and moreover the minimal cofactor is 2 instead of 4. Nevertheless, we will see in the next section that formulas for doubling and adding in the jacobian are slightly faster in the case of type Ib.

Concerning curves of type II, even if it was not obvious at first sight, we have the following properties on the cardinality of the jacobian:

Proposition 3. *Let C be a type II hyperelliptic curve of genus 2 defined over $\mathbb{F}_{2^n}$ by the equation*

$$y^2 + xy = x^5 + f_3x^3 + \varepsilon x^2 + f_0$$

The minimal cofactor is 4 if $\varepsilon = 0$ and 2 if $\varepsilon = 1$.

Sketch of the proof:
The divisor $\left(0, \sqrt{f_0}\right) + \infty$ is the only one divisor of order 2 and it exists a divisor D such that $2D = \left(0, \sqrt{f_0}\right) + \infty$ (i.e. a divisor of order 4) if and only if $\varepsilon = 0$. This proves that the cardinality of the jacobian is congruent to zero modulo 4 if $\varepsilon = 0$ and to 2 if $\varepsilon = 1$. $\square$

In this last case, we find many of both good curves and nice curves. From these results, it appears that, among hyperelliptic curves of genus 2 in characteristic 2, the curves of type II with $\varepsilon = 1$ are the best from a security point of view.

5 Application to jacobian scalar multiplication

We use this classification of hyperelliptic curve of genus 2, to rewrite and even optimize formulas of Lange for mixed addition and doubling on their jacobian. Lange uses three types of coordinates, affine [9], projective [10] and weighted projective [11]. In these papers Lange chose the coefficients of h in $\mathbb{F}_2$. In the last sections, we proved that we can't always assume that. That is why contrary to Lange we count multiplications by h_0 and h_1.

These formulas can be found in the appendix for curves of type II which is the most efficient. In fact the mixed addition formulas are just those of Lange rewritten in characteristic 2. We did not rewrite formulas for classical addition as they are also the same as Lange one's. Nevertheless, for doubling, our formulas are slightly different and optimized for each type of curve. Formulas for general cases and curves of type I can be found on the web page of the author.

Besides, we also introduced a new system of coordinates called *Modified Projective Coordinates*. Based on Projective representation, we add two coordinates Z_0, Z_1. So the septuple $[U_1, U_0, V_1, V_0, Z_0, Z_1, Z]$ stand for $[x^2 + U_1/Z + U_0/Z, x^2 + V_1/Z + V_0/Z]$ and $Z_0 = h_0 Z$, $Z_1 = h_1 Z$. The formulas for addition are the same as for projective one's but we gain some multiplications in doubling. The complexities we obtained are listed in the following table.

<table>
<tr><th></th><th>General case</th><th colspan="2">type I</th><th>type II</th></tr>
<tr><td>Affine</td><td></td><td colspan="2"></td><td></td></tr>
<tr><td>Addition</td><td>25M + I</td><td colspan="2">25M + I</td><td>24M + I</td></tr>
<tr><td>Doubling</td><td>27M + I</td><td colspan="2">26M + I</td><td>18M + I</td></tr>
<tr><td>Projective</td><td></td><td>(Ia)</td><td>(Ib)</td><td></td></tr>
<tr><td>Mixed Addition</td><td>45M</td><td>45M</td><td>44M</td><td>42M</td></tr>
<tr><td>Doubling</td><td>45M</td><td>44M</td><td>41M</td><td>31M</td></tr>
<tr><td>Modified Projective</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Mixed Addition</td><td>45M</td><td>45M</td><td>44M</td><td>42M</td></tr>
<tr><td>Doubling</td><td>43M</td><td>42M</td><td>40M</td><td>31M</td></tr>
<tr><td>Weighted Projective</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Mixed Addition</td><td>42M</td><td>42M</td><td>41M</td><td>40M</td></tr>
<tr><td>Doubling</td><td>46M</td><td>45M</td><td>42M</td><td>27M</td></tr>
</table>

We see we gain at least one multiplication in each system of coordinates for doubling, and of course more for each type of curve. The best performance was produced using type II.

We also noticed the weighted projective coordinates are only interesting for additions in the general case and type I. Thus, the use of projective and modified projective coordinates is more interesting if we use scalar multiplication methods such as sliding window (since it uses much more doubling than adding). Nevertheless, weighted projective coordinates are still competitive in type II or if one has to use doubling and adding at each step, for instance to resist against power analysis in restricted environments like smart cards. It can also be used with algorithms like BGMW, where doublings are pre-computed.
For example in type I, what we gain in addition by using weighted projective coordinates instead of modified projective, we lose in doubling.

Besides, one has to keep in mind that weighted projective coordinates uses up more memory, which has to be taken into account by anyone who wants to implement in restricted environments.

6 Conclusion

We studied genus 2 isomorphism classes of curves in characteristic 2. They are classified in three types. Type III curves are supersingular. We focused our effort on type I and type II and found optimal forms for these curves, just as the short Weierstrass form. For these types of curves we studied the security and the arithmetic on their jacobian.

In addition, we rewrote and optimized formulas of Lange in characteristic 2, and we introduced a new system of coordinate.

We noticed that both from the arithmetic and the security point of view, curves of the form

$$y^2 + xy = x^5 + f_3x^3 + x^2 + f_0$$

are the best for cryptographic use.
Hence we recommend this type for future standards.

Appendix: formulas for hyperelliptic curves over $\mathbb{F}_{2^n}$ of type II: $y^2 + xy = x^5 + f_3x^3 + \varepsilon x^2 + f_0$

Affine case

Affine Doubling with type II: $y^2 + xy = x^5 + f_3x^3 + \varepsilon x^2 + f_0$ with $\varepsilon \in \mathbb{F}_2$		
Input	$D = [u_1, u_0, v_1, v_0]$	
Output	$2D = [u'_1, u'_0, v'_1, v'_0]$	
Step	Operations	Cost
1	resultant r: $r = u_0$	
2	compute almost inverse: $1 = inv_1$, $u_1 = inv_0$	
3	compute k: $k_1 = u_1^2 + f_3$ $k_0 = u_1k_1 + v_1^2 + v_1 + \varepsilon$	2S, 1M
4	compute $s = kinv \bmod u$: Karatsuba is useless now $s_1 = k_0 + u_1k_1$ $s_0 = k_1u_0$ for $s_1 \neq 0$	1M
5	precomputation $t_0 = (u_0s_1)^{-1}$, $t_1 = u_0t_0$, $t_2 = s_1^2t_0$, $t_3 = u_0t_1$, $s_0 = s_0t_1$	1I, 1S, 5M
6	compute l $l_2 = u_1 + s_0$, $l_1 = u_1s_0 + u_0$, $l_0 = u_0s_0$	2M
7	compute u' $u'_0 = s_0^2 + t_3$ $u'_1 = t_3^2$	2S
8	compute v' $t_0 = u'_1(l_2 + u'_1) + u'_0 + l_1$ $v'_1 = t_2t_0 + v_1 + 1$ $t_0 = u'_0(l_2 + u'_1) + l_0$ $v'_0 = t_2t_0 + v_0$	4M
total		1I, 5S, 13M

Projective case

Projective Doubling with type II: $y^2 + xy = x^5 + f_3x^3 + \varepsilon x^2 + f_0$ with $\varepsilon \in \mathbb{F}_2$		
Input	$D = [U_1, U_0, V_1, V_0, Z]$	
Output	$2D = [U_1', U_0', V_1', V_0', Z']$	
Step	Operations	Cost
1	precomputation and resultant r: $t_0 = Z^2$, $t_1 = U_1^2$ $r = U_0Z$	2S, 1M
2	compute almost inverse: useless $inv_0 = U_1Z$, $inv_1 = Z$	
3	compute k: $k_1 = f_3t_0 + t_1$, $k_0 = U_1k_1 + Z(\varepsilon t_0 + V_1(Z + V_1))$	4M
4	compute $s = kinv \bmod u$: Karatsuba is useless now $t_2 = k_0U_1$ $s_1 = k_0Z$ $s_0 = k_1r + t_2$ for $s_1 \neq 0$	3M
5	precomputation and compute l $t_0 = t_0r$, $r = t_0s_1$, $t_1 = s_1k_0$, $t_3 = U_0k_0$ $l_2 = s_1t_2$, $l_0 = s_0t_3$, $l_1 = (t_2 + t_3)(s_0 + s_1) + l_2 + l_0$	7M
6	compute U' $U_0' = s_0^2 + r$ $U_1' = t_0^2$	2S
7	precomputation: $l_2 = l_2 + s_0s_1 + U_1'$, $s_1 = s_1^2$, $t_2 = rt_1$ $t_0 = U_0'l_2 + l_0s_1$, $t_1 = U_1'l_2 + s_1(U_0' + l_1)$	1S, 6M
8	adjust: $Z' = s_1r$, $U_1' = U_1'r$, $U_0' = U_0'r$	3M
9	compute V' $V_0' = t_0 + t_2V_0$ $V_1' = t_1 + t_2V_1 + Z'$	2M
total		5S, 26M

Modified projective coordinates, are obviously useless in this case.

Weighted projective case

Weighted projective Mixed Addition with type II: $y^2 + xy = x^5 + f_3x^3 + \varepsilon x^2 + f_0$ with $\varepsilon \in \mathbb{F}_2$		
Input	$D_1 = [u_{11}, u_{10}, v_{11}, v_{10}]$, $D_2 = [U_{21}, U_{20}, V_{21}, V_{20}, Z_{20}, Z_{21}, z_{20}, z_{21}, z_{22}, z_{23}]$	
Output	$D_1 + D_2 = [U'_1, U'_0, V'_1, V'_0, Z'_0, Z'_1, z'_0, z'_1, z'_2, z'_3]$	
Step	Operations	Cost
1	precomputation and resultant r: $t_1 = u_{11}z_{20} + U_{21}$, $t_2 = u_{10}z_{20} + U_{20}$, $t_0 = u_{11}t_1 + t_2$ $r = u_{10}t_1^2 + t_2t_0$, $t_3 = rz_{22}$, $Z'_1 = t_3Z_{20}$	1S, 7M
2	compute almost inverse: nothing to do $t_1 = inv_1$, $t_0 = inv_0$	
3	compute almost s: $t_4 = V_{10}z_{23} + V_{20}$, $t_5 = V_{11}z_{23} + V_{21}$, $s_0 = (t_2t_0) + u_{10}(t_3t_1)$ $s_1 = (t_0 + t_1)(t_4 + t_5) + (t_2t_0) + (t_3t_1)(1 + u_{11})$ for $s_1 \neq 0$	7M
4	precomputation: $Z'_0 = s_1Z_{20}$, $t_0 = rs_1$, $t_3 = t_3^2$, $t_4 = s_0Z_{20}$, $s_0 = s_0s_1$, $s_1 = s_1^2$, $z'_0 = {Z'_0}^2$, $z'_1 = {Z'_1}^2$, $z'_2 = Z'_0Z'_1$, $z'_3 = z'_0z'_2$	4S, 6M
5	compute l $l_2 = s_1u_{21}$, $l_0 = s_0u_{20}$, $l_1 = (s_0 + s_1)(u_{21} + u_{20}) + l_0 + l_2$	3M
6	compute U' $t_5 = t_1s_1$ $U'_0 = t_4^2 + u_{11}t_5 + s_1t_2 + z'_2 + t_1t_3$ $U'_1 = t_5 + z'_1$	1S, 3M
7	compute V' $t_1 = l_2 + Z'_0t_4 + U'_1$, $t_2 = t_1U'_0$, $t_3 = t_1U'_1$ $V'_1 = t_3 + z'_0(l_1 + t_0V_{21} + U'_0 + z'_2)$ $V'_0 = t_2 + z'_0(l_1 + t_0V_{20})$	8M
total		6S, 34M

Weighted projective Doubling with type II: $y^2+xy=x^5+f_3x^3+\varepsilon x^2+f_0$ with $\varepsilon \in \mathbb{F}_2$		
Input	$D=[U_1,U_0,V_1,V_0,Z_0,Z_1,z_0,z_1,z_2,z_3]$	
Output	$2D=[U_1',U_0',V_1',V_0',Z_0',Z_1',z_0',z_1',z_2',z_3']$	
Step	Operations	Cost
1	resultant r: $r=z_0U_0,\ t_0=rz_2,\ Z_1'=t_0z_2$	3M
2	compute almost inverse: useless $z_0=inv_1,\ z_0U_1=inv_0$	
3	compute k: $t_0=(\sqrt{f_3}z_0+U_1)^2$ with precomputation of $\sqrt{f_3}$ $k_1=t_0z_1$ $k_0=U_1k_1+V_1(V_1+z_3)+\varepsilon z_3^2$	2S, 4M
4	compute $s=kinv \bmod u$: Karatsuba is useless now $s_1=k_0$ $s_0=s_1U_1+k_1r$ for $s_1\neq 0$	2M
5	precomputation $Z_0'=s_1,\ t_0=t_1Z_0',\ r=s_0^2,\ s_0=s_0Z_0'$ $z_0'=Z_0'^2,\ z_1'=Z_1'^2,\ z_2'=Z_0'Z_1',\ z_3'=z_0'z_2'$	3S, 4M
6	compute l $l_2=U_1z_0',\ l_0=U_0s_0,\ l_1=(s_0+z_0')(U_1+U_0)+l_0+l_2$ $l_2=l_2+s_0$	3M
7	compute U' $U_0'=r+z_2'$ $U_1'=z_1'$	
8	compute V' $t_1=(l_2+U_1')U_0'$ $V_0'=t_1+z_0'(l_0+t_0V_0)$ $t_1=(l_2+U_1')U_1'$ $V_1'=t_1+z_0'(l_1+t_0V_1+U_0')+z_3'$	6M
total		5S, 22M

Remark: for the general case we choose the following form:

$$y^2+(h_2x^2+h_1x+h_0)y=x^5+f_4x^4+f_3x^3+f_2x^2+f_1x+f_0 \text{ with } h_2,f_4,f_3,f_2\in\mathbb{F}_2$$

as for type I, there is no f_2 or f_3 and $f_4\in\mathbb{F}_2$, and for type II $f_2\in\mathbb{F}_2$, there is no f_4 and following the remark of theorem 2 we can also erase f_3.
The formulas are mostly the same of T. Lange [9], [10], [11], but can be found in the web page of the author.

References

[1] D.G. Cantor. *Computing on the Jacobian of a hyperelliptic curve* Math. Comp., vol. 48, pp. 95-101, 1987.

[2] Y. Choie and D. Yun. *Isomorphism classes of hyperelliptic curves of genus 2 over* $\mathbb{F}_q$, in ACISP 2002. LNCS, vol. 2384, pp. 190-202, 2002.

[3] G. Frey and H. Rück. *A remark concerning m-divisibility and the dicrete logarithm in the divisor class group of curves,* Math. Comp., vol. 62, pp. 865-874, 1994.

[4] S. Galbraith. *Supersingular curves in cryptography,* in Advances in Cryptology Asiacrypt 2001, LNCS, vol. 2248, pp. 495-513, 2001.

[5] R. Harley. *Fast arithmetic on genus 2 curves.* available at `http://cristal.inria.fr/ harley/hyper`, 2000.

[6] N. Koblitz. *Elliptic Curves cryptosystem* Math. Comp., vol. 48, pp. 203-209, 1987.

[7] N. Koblitz. *Hyperelliptic cryptosystem* J. Crypto, vol. 1, pp. 139-150, 1989.

[8] N. Koblitz. *Algebraic aspects of cryptosystem* Springer, 1998.

[9] T. Lange. *Efficient Arithmetic on Genus 2 Hyperelliptic Curves over Finite Fields via Explicit Formulae.* Cryptology ePrint Archive, Report 2002/121, 2002. `http://eprint.iacr.org/`

[10] T. Lange. *Inversion-Free Arithmetic on Genus 2 Hyperelliptic Curves.* Cryptology ePrint Archive, Report 2002/147, 2002.

[11] T. Lange. *Weighted Coordinates on Genus 2 Hyperelliptic Curves.* Cryptology ePrint Archive, Report 2002/153, 2002.

[12] P. Lockhart. *On the discriminant of a hyperelliptic curve,* Trans. Ame. Math. Soc. 342, pp. 729-752, 1994.

[13] V. Miller. *Uses of Elliptic Curves in cryptography,* in Advances in Cryptology CRYPTO'85, LNCS, vol. 218, pp. 417-426, 1986.

[14] H.G. Rück. *On the discrete logarithms in the divisor of class group of curves,* Math. Comp., vol. 68, pp. 805-806, 1999.

[15] F. Vercauteren. *Computing Zeta Functions of Hyperelliptic Curves over Finite Fields of Characteristic 2,* in Advances in Cryptology CRYPTO'02, LNCS, vol. 2248, pp. 369-384, 2002.

[16] F. Zhang, S. Liu and K. Kim. *Compact representation of domain parameters of hyperelliptic curve cryptosystems,* in ACISP 2002. LNCS, vol. 2384, pp. 203-213, 2002.