

# Efficient Arithmetic on Binary Genus-2 Curves

Peter Birkner and Tanja Lange

Coding Theory and Cryptology, Department of Mathematics and Computer Science,  
Eindhoven University of Technology,  
P.O. Box 513, 5600 MB Eindhoven, The Netherlands  
p.birkner@tue.nl, tanja@hyperelliptic.org

## 1 Introduction

Many cryptographic protocols such as signature schemes base their security on the difficulty of the discrete-logarithm problem (DLP) in some group. For the efficiency of the protocol, it is important to choose secure groups in which the group operations are fast. Typically, addition of two group elements and scalar multiplication are needed

In this paper, we focus on the arithmetic in divisor-class groups of a certain family of hyperelliptic curves of genus 2 over binary fields. We give new explicit formulas to perform divisor-class addition and divisor-class doubling in different weighted projective coordinates. Affine formulas for these curves are due to Lange [Lan05] and Lange and Stevens [LS05]. However, on most platforms inversions are prohibitively expensive and so inversion-free systems are used. The preprint introducing recent coordinates [Lan] appeared only in the workshop handout and was not widely circulated. It contained a comparison table for different coordinate systems for 2-rank 1 curves but did not provide the formulas for systems other than the recent coordinates. The treatment of arithmetic on 2-rank 1 curves in [DL05] failed to give a comparison or a recommendation for this type of curves.  $s$  (see [Lan] and Section 14.5.5.b in [DL05]). In this paper, we first improve the doubling formulas in recent coordinates and then present a new coordinate system that is similar to Lange's new coordinates (cf. Section 6 in [Lan05] and Section 14.5.4.a in [DL05]) but more suited for this curve shape since it is faster for doublings. Recent coordinates are weighted projective coordinates with weights  $(1, 2)$ . Our new coordinate system has weights  $(2, 3)$ . These two weights are the natural choices for this type of curve and we show that the recent coordinates lead to faster arithmetic.

## 2 Choice of curve

In this paper, we work with hyperelliptic curves of Type II (cf. Section 14.5.1 in [DL05]) over  $\mathbb{F}_{2^d}$  for an odd integer  $d$ , i.e. the curve equation is of the form

$$C : y^2 + xy = x^5 + f_3x^3 + f_2x^2 + f_0, \quad (1)$$

where  $f_2 \in \mathbb{F}_2$  and  $f_3, f_0 \in \mathbb{F}_{2^d}$ . If the linear term in  $y$  also has degree 1 in  $x$ , one can generically obtain this form by isomorphic transformations and does hence not lose any generality. Curves of this form have 2-rank 1, i.e. the 2-torsion subgroup of the divisor-class group is isomorphic to  $\mathbb{Z}/2\mathbb{Z}$ . We have chosen this

type of curve because the explicit addition and doubling formulas give better performance over curves with 2-rank 2. Note that curves with 2-rank 0 are supersingular and therefore weak under the Frey-Rück attack and hence not interesting for DLP-based cryptosystems.

Gaudry and Lubicz [GL09] consider scalar multiplication on the Kummer surface of 2-rank 2 curves and obtain formulas for which the combination of one doubling and one differential addition (addition of  $\overline{D}_1$  and  $\overline{D}_2$  given the difference  $\overline{D}_1 - \overline{D}_2$ ) takes only 15M (multiplications), 9S (squarings) and 3D (multiplications by curve constants). This is faster than a doubling on our curves and thus for applications, like Diffie-Hellman key exchange which only need scalar multiplication, their arithmetic is preferable. However, the lack of (non-differential) addition means that their system cannot be used in, e.g. signature schemes. This is also nicely reflected by the speeds in the eBATS competition (part of eBACS [eba]). The HECTOR bat is based on 2-rank 1 curves and uses recent coordinates. For signatures it is faster than any of the competitors.

### 3 Explicit formulas in recent coordinates

In 2005, Lange ([Lan], see also Section 14.5.5.b in [DL05]) suggested to use  $[U_1, U_0, V_1, V_0, Z, z]$  to represent the divisor class

$$[x^2 + (U_1/Z)x + U_0/Z, (V_1/Z^2)x + V_0/Z^2] \quad (2)$$

and  $z = Z^2$ . This representation is called *recent coordinates*. The weights of the coordinates are (1, 2), meaning that the first two elements are divided by  $Z$  while the last two are divided by  $Z^2$ . In the following, we present new, faster doubling formulas in recent coordinates.

---

**Algorithm 1** (DBL22,  $h(x) = x, f(x) = x^5 + f_3x^3 + f_2x^2 + f_0$ )

---

INPUT: A divisor class  $\overline{D} = [U_1, U_0, V_1, V_0, Z, z]$  in recent coordinates

OUTPUT: The doubled divisor class  $[U'_1, U'_0, V'_1, V'_0, Z', z'] = [2]\overline{D}$

---

- 1:  $Z_4 \leftarrow z^2, t_1 \leftarrow f_0Z_4 + V_0^2, t_2 \leftarrow U_1^2 + f_3z$  ▷ 3S+2D
  - 2:  $a_1 \leftarrow U_0^2, a_2 \leftarrow a_1Z, a_3 \leftarrow a_1Z_4, a_4 \leftarrow t_1a_3$  ▷ 3M+1S
  - 3:  $q_1 \leftarrow (t_2a_2 + U_1t_1)^2 + a_4$  ▷ 2M+1S
  - 4:  $q_2 \leftarrow t_1^2, q_3 \leftarrow q_2^2, q_4 \leftarrow a_1z, q_5 \leftarrow t_1t_2, q_6 \leftarrow (a_3 + q_5)t_1$  ▷ 3M+2S
  - 5:  $U'_0 \leftarrow q_1, U'_1 \leftarrow a_3q_4, V'_0 \leftarrow q_6q_1 + q_3q_4$  ▷ 3M
  - 6:  $V'_1 \leftarrow q_4(q_5q_6 + a_3a_4) + q_3(f_2Z_4 + V_1^2), Z' \leftarrow q_2z, z' \leftarrow Z'^2$  ▷ 5M+2S+1D
  - 7: **return**  $[U'_1, U'_0, V'_1, V'_0, Z', z']$  ▷ Total: 16M+9S+3D
- 

The following addition formulas in recent coordinates for curves of Type II are taken from Algorithm 14.50 in [DL05]. The input can be either two divisor classes in recent coordinates, or one divisor class in recent coordinates and the other one in affine coordinates. In the latter case, let  $Z_1 = z_1 = 1$ . An addition in this way is called *mixed addition*; the different operation counts are stated in parentheses.

---

**Algorithm 2** (ADD,  $h(x) = x$ ,  $f(x) = x^5 + f_3x^3 + f_2x^2 + f_0$ )

INPUT: Two divisor classes  $\overline{D}_1 = [U_{11}, U_{10}, V_{11}, V_{10}, Z_1, z_1]$  and  
 $\overline{D}_2 = [U_{21}, U_{20}, V_{21}, V_{20}, Z_2, z_2]$

OUTPUT: The divisor class  $[U'_1, U'_0, V'_1, V'_0, Z', z'] = \overline{D}_1 \oplus \overline{D}_2$

---

- 1:  $Z \leftarrow Z_1 Z_2, z \leftarrow Z^2, \tilde{U}_{21} \leftarrow U_{21} Z_1, \tilde{U}_{20} \leftarrow U_{20} Z_1$  ▷ 3M+1S (-)
- 2:  $\tilde{V}_{21} \leftarrow V_{21} z_1, \tilde{V}_{20} \leftarrow V_{20} z_1$  ▷ 2M (-)
- 3:  $y_1 \leftarrow U_{11} Z_2 + \tilde{U}_{21}, y_2 \leftarrow U_{10} Z_2 + \tilde{U}_{20}$  ▷ 2M
- 4:  $y_3 \leftarrow U_{11} y_1 + y_2 Z_1, r \leftarrow y_2 y_3 + y_1^2 U_{10}$  ▷ 4M+1S (3M+1S)
- 5:  $w_0 \leftarrow V_{10} z_2 + \tilde{V}_{20}, w_1 \leftarrow V_{11} z_2 + \tilde{V}_{21}$  ▷ 2M
- 6:  $w_2 \leftarrow y_3 w_0, w_3 \leftarrow y_1 w_1$  ▷ 2M
- 7:  $s_1 \leftarrow (y_3 + y_1 Z_1)(w_0 + w_1) + w_2 + w_3(Z_1 + U_{11})$  ▷ 3M (2M)
- 8:  $s_0 \leftarrow w_2 + U_{10} w_3$  ▷ 1M
- 9:  $\overline{Z} \leftarrow s_1 r, w_4 \leftarrow r Z, w_5 \leftarrow w_4^2, S \leftarrow s_0 Z, Z' \leftarrow Z \overline{Z}$  ▷ 4M+1S
- 10:  $\tilde{s}_0 \leftarrow s_0 Z', \tilde{s}_1 \leftarrow s_1 \overline{Z}, \tilde{s}_1 \leftarrow \tilde{s}_1 Z$  ▷ 3M
- 11:  $L_2 \leftarrow \tilde{s}_1 \tilde{U}_{21}, \ell_2 \leftarrow L_2 Z, \ell_0 \leftarrow \tilde{s}_0 \tilde{U}_{20}$  ▷ 3M
- 12:  $\ell_1 \leftarrow (\tilde{U}_{21} + \tilde{U}_{20})(\tilde{s}_0 + \tilde{s}_1) + \ell_2 + \ell_0, \ell_2 \leftarrow L_2 + \tilde{s}_0$  ▷ 1M
- 13:  $U'_0 \leftarrow r(S^2 + y_1(s_1^2(y_1 + \tilde{U}_{21}) + Z w_5) + z Z') + y_2 \tilde{s}_1$  ▷ 6M+2S
- 14:  $U'_1 \leftarrow y_1 \tilde{s}_1 + w_4 w_5$  ▷ 2M
- 15:  $w_1 \leftarrow \ell_2 + U'_1, U'_1 \leftarrow U'_1 w_4, \overline{Z} \leftarrow Z' \overline{Z}, \ell_0 \leftarrow \ell_0 \overline{Z}$  ▷ 3M
- 16:  $w_2 \leftarrow U'_1 w_1 + (U'_0 + \ell_1) \overline{Z}, \overline{Z} \leftarrow \overline{Z}^2$  ▷ 2M+1S
- 17:  $V'_1 \leftarrow w_2 s_1 + (\tilde{V}_{21} + z) \overline{Z}, U'_0 \leftarrow U'_0 r, w_2 \leftarrow U'_0 w_1 + \ell_0$  ▷ 4M
- 18:  $V'_0 \leftarrow w_2 s_1 + \tilde{V}_{20} \overline{Z}, Z' \leftarrow Z'^2, z' \leftarrow Z'^2$  ▷ 2M+2S
- 19: **return**  $[U'_1, U'_0, V'_1, V'_0, Z', z']$  ▷ Total: 49M+8S

▷ Mixed addition ( $Z_1 = z_1 = 1$ ): 42M+7S

---

Note that addition benefits by 1S from having both inputs and the output with the sixth value  $z_i$ . If it is important to use as little memory as possible, then those values can be omitted. For doublings and mixed additions, using  $z_i$  does not provide any advantage and should thus be skipped to save space.

#### 4 Arithmetic in weighted coordinates with weights (2, 3)

In this section, we propose a new coordinate system that is similar to Lange's new coordinates (see Section 6 in [Lan05] and Section 14.5.4.a in [DL05]). Let the 7-tuple  $[U_1, U_0, V_1, V_0, Z, z, v]$  represent the divisor class

$$[x^2 + (U_1/Z^2)x + U_0/Z^2, (V_1/Z^3)x + V_0/Z^3] \quad (3)$$

and let  $z = Z^2, v = Z^3$ . The weights of these coordinates are (2, 3) according to the exponents of  $Z$ . The formulas for divisor-class doubling and divisor-class addition are as follows:

---

**Algorithm 3** (DBL,  $h(x) = x$ ,  $f(x) = x^5 + f_3x^3 + f_2x^2 + f_0$ ,  $b = \sqrt{f_3}$ )

---

INPUT: A divisor class  $\overline{D} = [U_1, U_0, V_1, V_0, Z, z, v]$

OUTPUT: The doubled divisor class  $[U'_1, U'_0, V'_1, V'_0, Z', z', v'] = [2]\overline{D}$

---

- 1:  $z_0 \leftarrow U_0^2$ ,  $k \leftarrow (U_1 + bz)^2$  ▷ 2S+1D
  - 2:  $z_6 \leftarrow v^2$ ,  $w_0 \leftarrow f_0z_6 + V_0^2$ ,  $Z' \leftarrow w_0v$ ,  $z_1 \leftarrow kz_0$ , ▷ 2M+2S+1D
  - 3:  $s_0 \leftarrow z_1 + U_1w_0$ ,  $z_5 \leftarrow zv$ ,  $z_4 \leftarrow z_0z_5$ ,  $U'_1 \leftarrow z_4^2$  ▷ 3M+1S
  - 4:  $U''_0 \leftarrow (s_0^2 + z_0Z'v)Z$ ,  $z' \leftarrow Z'^2$ ,  $v' \leftarrow Z'z'$  ▷ 4M+2S
  - 5:  $w_3 \leftarrow z_0z_6 + kw_0$ ,  $W_0 \leftarrow w_0^2Z'$  ▷ 3M+1S
  - 6:  $V'_1 \leftarrow w_3z_1Z' + z_4U'_1 + f_2v' + V_1^2W_0$  ▷ 4M+1S+1D
  - 7:  $V'_0 \leftarrow w_3U''_0 + z_0zW_0$ ,  $U'_0 \leftarrow U''_0Z$  ▷ 4M
  - 8: **return**  $[U'_1, U'_0, V'_1, V'_0, Z', z', v']$  ▷ Total: 20M+9S+3D
- 

In the addition algorithm, we again state the operation counts for mixed addition (i.e.  $Z_1 = z_1 = v_1 = 1$ ) in parentheses.

---

**Algorithm 4** (ADD,  $h(x) = x$ ,  $f(x) = x^5 + f_3x^3 + f_2x^2 + f_0$ )

---

INPUT: Two divisor classes  $\overline{D}_1 = [U_{11}, U_{10}, V_{11}, V_{10}, Z_1, z_1, v_1]$  and  $\overline{D}_2 = [U_{21}, U_{20}, V_{21}, V_{20}, Z_2, z_2, v_2]$

OUTPUT: The divisor class  $[U'_1, U'_0, V'_1, V'_0, Z', z', v'] = \overline{D}_1 \oplus \overline{D}_2$

---

- 1:  $\tilde{U}_{21} \leftarrow U_{21}z_1$ ,  $\tilde{U}_{20} \leftarrow U_{20}z_1$ ,  $\tilde{V}_{21} \leftarrow V_{21}v_1$  ▷ 3M (-)
- 2:  $\tilde{V}_{20} \leftarrow V_{20}v_1$ ,  $Z_3 \leftarrow Z_1Z_2$ ,  $W_1 \leftarrow Z_3^2$  ▷ 2M+1S (-)
- 3:  $y_1 \leftarrow U_{11}z_2 + \tilde{U}_{21}$ ,  $y_2 \leftarrow U_{10}z_2 + \tilde{U}_{20}$  ▷ 2M
- 4:  $y_3 \leftarrow U_{11}y_1 + y_2z_1$ ,  $r \leftarrow y_2y_3 + y_1^2U_{10}$ ,  $\tilde{Z}_2 \leftarrow rZ_3$  ▷ 5M+1S (4M+1S)
- 5:  $Z'_2 \leftarrow \tilde{Z}_2W_1$ ,  $\tilde{Z}_2 \leftarrow \tilde{Z}_2^2$ ,  $\tilde{Z}_2 \leftarrow \tilde{Z}_2W_1$  ▷ 2M+1S
- 6:  $w_0 \leftarrow V_{10}v_2 + \tilde{V}_{20}$ ,  $w_1 \leftarrow V_{11}v_2 + \tilde{V}_{21}$  ▷ 2M
- 7:  $w_2 \leftarrow y_3w_0$ ,  $w_3 \leftarrow y_1w_1$  ▷ 2M
- 8:  $s_1 \leftarrow (y_3 + z_1y_1)(w_0 + w_1) + w_2 + w_3(z_1 + U_{11})$  ▷ 3M (2M)
- 9:  $s_0 \leftarrow w_2 + U_{10}w_3$ ,  $S_1 \leftarrow s_1^2$ ,  $Z'_1 \leftarrow s_1W_1$  ▷ 2M+1S
- 10:  $R \leftarrow rZ'_1$ ,  $S_0 \leftarrow s_0W_1$ ,  $S \leftarrow S_0Z'_1$ ,  $S_0 \leftarrow S_0^2$  ▷ 3M+1S
- 11:  $z'_1 \leftarrow Z_1'^2$ ,  $z'_2 \leftarrow Z_2'^2$ ,  $z'_3 \leftarrow Z_1'Z_2'$ ,  $z'_4 \leftarrow z'_1z'_3$  ▷ 2M+2S
- 12:  $s_1 \leftarrow s_1Z'_1$ ,  $s_0 \leftarrow s_0Z'_1$ ,  $\ell_2 \leftarrow s_1\tilde{U}_{21}$ ,  $\ell_0 \leftarrow s_0\tilde{U}_{20}$  ▷ 4M
- 13:  $\ell_1 \leftarrow (s_0 + s_1)(\tilde{U}_{20} + \tilde{U}_{21}) + \ell_0 + \ell_2$ ,  $\ell_2 \leftarrow \ell_2 + S$  ▷ 1M
- 14:  $U'_0 \leftarrow S_0 + y_1(S_1(y_1 + \tilde{U}_{21}) + \tilde{Z}_2) + y_2s_1 + z'_3$  ▷ 3M
- 15:  $U'_1 \leftarrow y_1s_1 + z'_2$ ,  $\ell_2 \leftarrow \ell_2 + U'_1$ ,  $w_0 \leftarrow \ell_2U'_0$  ▷ 2M
- 16:  $w_1 \leftarrow \ell_2U'_1$ ,  $V'_1 \leftarrow w_1 + z'_1(\ell_1 + R\tilde{V}_{21} + U'_0 + z'_3)$  ▷ 3M
- 17:  $V'_0 \leftarrow w_0 + z'_1(\ell_0 + R\tilde{V}_{20})$ ,  $z'_2 \leftarrow Z_2'^2$ ,  $U'_1 \leftarrow U'_1z'_2$  ▷ 3M+1S

18:  $U'_0 \leftarrow U'_0 z'_2, V'_1 \leftarrow V'_1 z'_2, V'_0 \leftarrow V'_0 z'_2, Z' \leftarrow Z'_1 Z'_2$  ▷ 4M  
19:  $z' \leftarrow Z'^2, v' \leftarrow z' v'$  ▷ 1M+1S  
20: **return**  $[U'_1, U'_0, V'_1, V'_0, Z', z', v']$  ▷ Total: 49M+9S  
▷ Mixed addition ( $Z_1 = z_1 = v_1 = 1$ ): 42M+8S

---

## 5 Comparison

The following table gives the operation counts for divisor-class addition, divisor-class doubling and divisor-class mixed addition on hyperelliptic curves of the form (1) over binary fields. Note that [Lan] does not distinguish between M and D; the formulas use 3 multiplications by curve constants.

	Recent coords [Lan]	Recent coords This work	(2,3)-coords This work
DBL	17M+10S+3D	16M+9S+3D	20M+9S+3D
ADD	49M+8S	49M+8S	49M+9S
mADD	42M+7S	42M+7S	42M+8S

We improved doublings by 1M at the expense of 1S in recent coordinates. In binary fields squarings are significantly cheaper than multiplications, so this is a worthwhile tradeoff. The study of (2,3) coordinates arose out of adapting new coordinates to the faster doubling formulas [LS05] of 2-rank 1 curves and the observation, that the second  $Z$ -coordinate was not used there at all. This led to the hope that pure (2,3) coordinates, i.e. with only 1 additional coordinate, could improve the speed of addition. However, the formulas for this case are even slower than those in recent coordinates which confirms that recent coordinates are really the right choice for curves of form (1). A comparison with the results in [Lan05] shows that these curves are significantly faster than 2-rank 2 curves and thus that applications of hyperelliptic curves that require additions should choose recent coordinates for the implementation.

## References

- [ACD<sup>+</sup>05] R. Avanzi, H. Cohen, C. Doche, G. Frey, T. Lange, K. Nguyen, and F. Vercauteren. *The Handbook of Elliptic and Hyperelliptic Curve Cryptography*. CRC Press, 2005.
- [BL] D. J. Bernstein and T. Lange. eBACS: ECRYPT Benchmarking of Cryptographic Systems. URL: <http://bench.cr.yp.to>.
- [DL05] S. Duquesne and T. Lange. *Arithmetic of Hyperelliptic Curves*, chapter 14 in [ACD<sup>+</sup>05], pages 303–353. CRC Press, 2005.
- [GL09] P. Gaudry and D. Lubicz. The arithmetic of characteristic 2 Kummer surfaces and of elliptic Kummer lines. *Finite Fields and Their Applications*, 15(2):246–260, 2009.
- [Lan] T. Lange. Arithmetic on Binary Genus 2 Curves Suitable for Small Devices. Workshop on RFID and Lightweight Crypto, Graz, July 14-15, 2005.
- [Lan05] T. Lange. Formulae for Arithmetic on Genus 2 Hyperelliptic Curves. *Applicable Algebra in Engineering, Communication and Computing*, 15(5):295–328, 2005.
- [LS05] Tanja Lange and Marc Stevens. Efficient Doubling for Genus Two Curves over Binary Fields. In *Selected Areas in Cryptography – SAC 2004*, volume 3357 of *Lecture Notes in Computer Science*, pages 170–181. Springer-Verlag, 2005.