# Faster Group Operations on Special Elliptic Curves

Huseyin Hisil, Gary Carter, Ed Dawson
Information Security Institute, Queensland University of Technology,
{h.hisil, g.carter, e.dawson}@qut.edu.au

**Abstract**

This paper is on efficient implementation techniques of Elliptic Curve Cryptography. We improve group operation timings[1] for Hessian and Jacobi-intersection forms of elliptic curves. In this study, traditional coordinates of these forms are modified to speed up the addition operations. For the completeness of our study, we also recall the modified Jacobi-quartic coordinates which benefits from similar optimizations. The operation counts on the modified coordinates of these forms are as follows:

- Modified Hessian: Doubling 3M+6S, readdition 6M+6S, mixed addition 5M+6S, addition 6M+6S.
- Modified Jacobi-intersection: Doubling 2M+5S+1D, readdition 11M+1S+2D, mixed addition 10M+1S+2D, addition 11M+1S+2D.
- Modified Jacobi-quartic: Doubling 3M+4S, readdition 8M+3S+1D, mixed addition 7M+3S+1D, addition 8M+3S+1D.

We compare various elliptic curve representations with respect to their performance evaluations for different point multiplication algorithms. We note that Jacobi-quartics can provide the fastest timings for some S/M and D/M values in fast point multiplication implementations. We also show that Hessian form can provide the fastest timings for some S/M and D/M values when side-channel resistance is required for point multiplication.

Keywords: Elliptic curve arithmetic, unified addition, side channel attacks.

## 1 Introduction

One of the main challenges in elliptic curve cryptography is to perform scalar multiplication efficiently under different environment constraints (such as resistance for side channel attacks, bandwidth efficiency, less memory requirement). In the last decade, much effort has been spent in representing the elliptic curves in special forms which permit faster point doubling and addition. In particular,

[1] **M:** Field multiplication, **S:** Field squaring, **D:** Multiplication by a curve constant.

- Chudnovsky and Chudnovsky [1] reported the operation counts for inversion-free addition and doubling operations for Weierstrass, Hessian, Jacobi-intersection and Jacobi-quartic forms. Cohen, Miyaji and Ono [2] showed a faster method in Weierstrass form on Jacobian coordinates. Doche, Icart and Kohel [3] introduced the fastest doubling[2] and tripling in Weierstrass form on two different families of curves.

- Joye and Quisquater [6], Liardet and Smart [7], Brier and Joye [8], Billet and Joye [9] showed ways of doing point multiplication to resist side channel attacks on Hessian, Jacobi-intersection, Weierstrass and Jacobi-quartic forms, respectively in chronological order.

- Duquesne [10] improved the operation count for the Jacobi-quartic unified addition formula by extending the traditional coordinates. Here, "unified addition formula" means a point addition formula that can be used for point doubling.

- Edwards [11] introduced a new representation of elliptic curves. Bernstein and Lange [12, 13] showed the importance of this new form for providing fast arithmetic and side channel resistance. They have also built a database [4] of explicit formulae that are reported in the literature together with their own optimizations. Later, Bernstein and Lange [12, 13] introduced the inverted Edwards coordinates which improve the unified addition timings for Edwards curves.

This paper is a continuation of an earlier work by the authors in [14]. We focus our attention on the Hessian and the Jacobi-intersection forms. Our aim is to determine whether curve arithmetic can be improved for these forms. The improvements on the addition formulae are obtained from an approach analogous to that of Duquesne's [10]. The fast doubling formulae are obtained by choosing alternative polynomial expressions for the coordinates involved in the doubling formulae. This paper is organized as follows. We provide better operation counts for Hessian form in Section 2 and for Jacobi-intersection form in Section 3. We recall some previous results for Jacobi-quartic form [10, 4, 14] in Section 4. We provide comparisons of various systems and draw our conclusions in Section 5.

## 2 Modified Hessian Coordinates

The historical track of the explicit formulae for Hessian form is reported by Chudnovsky and Chudnovsky [1]. Let $K$ be a field with $\mathrm{char}(K) > 3$. For these fields, an elliptic curve in Hessian form is defined by $E(K)\colon x^3 + y^3 + 1 = 3dxy$ where $d^3 \neq 1$. The identity element is the point at infinity. (See [15, 6, 14]). The inversion-free addition formula is defined as follows. The inputs that set

[2] With the improvements of Bernstein, Birkner, Lange and Peters [4, 5].

$Z_3 = 0$ should be handled separately.

$$\begin{aligned} X_3 &= Y_1^2 X_2 Z_2 - X_1 Z_1 Y_2^2 \\ Y_3 &= X_1^2 Y_2 Z_2 - Y_1 Z_1 X_2^2 \\ Z_3 &= Z_1^2 X_2 Y_2 - X_1 Y_1 Z_2^2 \end{aligned}$$

We modify the traditional Hessian coordinates by appending new values denoted by the letters $R, S, T, U, V, W$. Two points;

$$(X_1 : Y_1 : Z_1 : R_1 : S_1 : T_1 : U_1 : V_1 : W_1), \quad (X_2 : Y_2 : Z_2 : R_2 : S_2 : T_2 : U_2 : V_2 : W_2)$$

with

$$R_1 \leftarrow X_1^2,\ S_1 \leftarrow Y_1^2,\ T_1 \leftarrow Z_1^2,\ U_1 \leftarrow 2X_1Y_1,\ V_1 \leftarrow 2X_1Z_1,\ W_1 \leftarrow 2Y_1Z_1,$$

$$R_2 \leftarrow X_2^2,\ S_2 \leftarrow Y_2^2,\ T_2 \leftarrow Z_2^2,\ U_2 \leftarrow 2X_2Y_2,\ V_2 \leftarrow 2X_2Z_2,\ W_2 \leftarrow 2Y_2Z_2$$

can be added as follows.

$$X_3 \leftarrow S_1V_2 - V_1S_2, \quad Y_3 \leftarrow R_1W_2 - W_1R_2, \quad Z_3 \leftarrow T_1U_2 - U_1T_2,$$

$$R_3 \leftarrow X_3^2, \quad S_3 \leftarrow Y_3^2, \quad T_3 \leftarrow Z_3^2, \quad U_3 \leftarrow (X_3 + Y_3)^2 - R_3 - S_3,$$
$$V_3 \leftarrow (X_3 + Z_3)^2 - R_3 - T_3, \quad W_3 \leftarrow (Y_3 + Z_3)^2 - S_3 - T_3.$$

The addition for modified Hessian coordinates costs **6M+6S**. This improves on the 12M figure reported in [1] at the cost of some more space requirement. The mixed addition can be derived by setting $Z_2 = 1$ and storing $R_2, S_2, U_2$. It costs **5M+6S**. Unlike other special forms, the addition formula does not depend on the curve parameter. This is certainly an advantage of the Hessian form. If side channel resistance is necessary, point doubling can be performed as

$$(Z_1 : X_1 : Y_1 : T_1 : R_1 : S_1 : V_1 : W_1 : U_1) + (Y_1 : Z_1 : X_1 : S_1 : T_1 : R_1 : W_1 : U_1 : V_1)$$

using the addition formula on modified Hessian coordinates. This strategy originates from Joye and Quisquater [6, p.6].

For speed oriented implementations, there is a perfectly fitting doubling strategy which requires no additional effort for generating the new coordinates. The traditional doubling formula can be expressed as follows. Note, this formula is essentially the same as reported in [1, p.425] (with each coordinate multiplied by 4).

$$\begin{aligned} X_3 &= (2X_1Y_1 - 2Y_1Z_1)(2X_1Z_1 + 2(X_1^2 + Z_1^2)) \\ Y_3 &= (2X_1Z_1 - 2X_1Y_1)(2Y_1Z_1 + 2(Y_1^2 + Z_1^2)) \\ Z_3 &= (2Y_1Z_1 - 2X_1Z_1)(2X_1Y_1 + 2(X_1^2 + Y_1^2)) \end{aligned}$$

The point $(X_1 : Y_1 : Z_1 : R_1 : S_1 : T_1 : U_1 : V_1 : W_1)$ can be added to itself as follows.

$$X_3 \leftarrow (U_1 - W_1)(V_1 + 2(R_1 + T_1)),$$

$$Y_3 \leftarrow (V_1 - U_1)(W_1 + 2(S_1 + T_1)),$$
$$Z_3 \leftarrow (W_1 - V_1)(U_1 + 2(R_1 + S_1)),$$
$$R_3 \leftarrow X_3^2, \quad S_3 \leftarrow Y_3^2, \quad T_3 \leftarrow Z_3^2, \quad U_3 \leftarrow (X_3 + Y_3)^2 - R_3 - S_3,$$
$$V_3 \leftarrow (X_3 + Z_3)^2 - R_3 - T_3, \quad W_3 \leftarrow (Y_3 + Z_3)^2 - S_3 - T_3.$$

This formula costs **3M+6S**. (See [14] for a 7M+1S doubling on the traditional coordinates). Like the addition formula, the doubling formula also does not depend on the curve parameter.

One can reduce the number of coordinates without storing $V, W$ and $U$. This does not effect doubling, readdition and mixed addition costs. However, the addition costs **6M+9S** in this case.

We comment that it is possible to derive unified addition formulae which does not require any permutations of the coordinates for performing doubling. One of such formulae is as follows. (The derivation of the new formula is given in Appendix-A).

$$\begin{aligned} X_3 &= X_1X_2(X_1Y_1Z_2^2 + X_2Y_2Z_1^2 - 3dX_1Y_1X_2Y_2) + (Y_1Y_2)^2Z_1Z_2 \\ Y_3 &= -Y_1Y_2(X_1Y_1Z_2^2 + X_2Y_2Z_1^2 - 3dX_1Y_1X_2Y_2) - (X_1X_2)^2Z_1Z_2 \\ Z_3 &= (X_1X_2)^3 - (Y_1Y_2)^3 \end{aligned}$$

We again use a modified coordinate system rather than the standard coordinates. Two points;

$$(X_1 : Y_1 : Z_1 : V_1 : W_1), \quad (X_2 : Y_2 : Z_2 : V_2 : W_2)$$

with

$$V_1 \leftarrow X_1Y_1, \quad W_1 \leftarrow Z_1^2, \quad V_2 \leftarrow X_2Y_2, \quad W_2 \leftarrow Z_2^2$$

can be added as follows.

$$A \leftarrow X_1X_2, \quad B \leftarrow Y_1Y_2, \quad C \leftarrow ((Z_1 + Z_2)^2 - W_1 - W_2)/2,$$
$$D \leftarrow A^2, \quad E \leftarrow B^2, \quad F \leftarrow D + E, \quad G \leftarrow ((A + B)^2 - F)/2,$$
$$H \leftarrow (V_1 + W_1)(V_2 + W_2) - (3d + 1)G - C^2,$$
$$X_3 \leftarrow AH + EC, \quad Y_3 \leftarrow -BH - DC, \quad Z_3 \leftarrow (A - B)(G + F),$$
$$V_3 \leftarrow X_3Y_3, \quad W_3 \leftarrow Z_3^2$$

This strategy costs **9M+6S+1D** which is faster than the unified addition in Weierstrass form in [8, 4]. However, it is slower than all other unified additions considered in this paper.

# 3 Modified Jacobi-intersection Coordinates

Let $K$ be a field with $\mathrm{char}(K) \neq 2, 3$ and let $a \in K$ with $a(1-a) \neq 0$. An elliptic curve in Jacobi-intersection form [1, 7, 14] is the set of points which satisfy the equations $s^2 + c^2 = 1$ and $as^2 + d^2 = 1$ simultaneously. The identity element is the point $(0, 1, 1)$. The inversion-free addition formula is defined as follows. The inputs that set $T_3 = 0$ should be handled separately.

$$\begin{aligned} S_3 &= S_1T_1C_2D_2 + C_1D_1S_2T_2 \\ C_3 &= C_1T_1C_2T_2 - S_1D_1S_2D_2 \\ D_3 &= D_1T_1D_2T_2 - aS_1C_1S_2C_2 \\ T_3 &= T_1^2C_2^2 + D_1^2S_2^2 \end{aligned}$$

We modify the traditional Jacobi-intersection coordinates by appending new values denoted by the letters $U$ and $V$. Two points;

$$(S_1 : C_1 : D_1 : T_1 : U_1 : V_1), \quad (S_2 : C_2 : D_2 : T_2 : U_2 : V_2)$$

with

$$U_1 \leftarrow S_1C_1, \quad V_1 \leftarrow D_1T_1, \quad U_2 \leftarrow S_2C_2, \quad V_2 \leftarrow D_2T_2$$

are added as follows.

$$E \leftarrow S_1D_2, \quad F \leftarrow C_1T_2, \quad G \leftarrow D_1S_2, \quad H \leftarrow T_1C_2, \quad J \leftarrow U_1V_2, \quad K \leftarrow V_1U_2,$$

$$S_3 \leftarrow (H+F)(E+G) - (J+K), \quad C_3 \leftarrow (H+E)(F-G) - (J-K),$$

$$D_3 \leftarrow (V_1 - aU_1)(U_2 + V_2) + aJ - K, \quad T_3 \leftarrow (H+G)^2 - 2K,$$

$$U_3 \leftarrow S_3C_3, \quad V_3 \leftarrow D_3T_3.$$

This formula costs **11M+1S+2D** which improves on the 13M+2S+1D reported in [7, 4]. The mixed addition can be derived by setting $T_2 = 1$ and storing $U_2$. It costs **10M+1S+2D** as opposed to the previously reported 11M+2S+1D in [7, 4] and 10M+2S+1D in [14, p.146].

The compatible doubling formula for this addition is different to the doubling formula explained in [1, 7]. The new doubling formula is as follows. (The derivation of the new formula is given in Appendix-B).

$$\begin{aligned} S_3 &= 2S_1C_1D_1T_1 \\ C_3 &= -D_1^2T_1^2 - aS_1^2C_1^2 + 2(S_1^2C_1^2 + C_1^4) \\ D_3 &= D_1^2T_1^2 - aS_1^2C_1^2 \\ T_3 &= D_1^2T_1^2 + aS_1^2C_1^2 \end{aligned}$$

The point, $(S_1 : C_1 : D_1 : T_1 : U_1 : V_1)$ can be added to itself as follows.

$$E \leftarrow V_1^2, \quad F \leftarrow U_1^2, \quad G \leftarrow aF, \quad T_3 \leftarrow E + G, \quad D_3 \leftarrow E - G,$$

$$C_3 \leftarrow 2(F + C_1^4) - T_3, \quad S_3 \leftarrow (U_1 + V_1)^2 - E - F, \quad U_3 \leftarrow C_3S_3, \quad V_3 \leftarrow D_3T_3.$$

This formula costs **2M+5S+1D**. The operation count is also the same on the traditional coordinates however it is only preferable when the curve constant is small. (i.e. D/M$\approx$0).

# 4 Modified Jacobi-quartic Coordinates

Let $K$ be a field with $\mathrm{char}(K) \neq 2, 3$. An elliptic curve in Jacobi-quartic form [9] is defined by $E(K)\colon y^2 = x^4 + 2ax^2 + 1$ where $a^2 - 1$ is nonzero. The identity element is the point $(0, 1)$. The leading study for the modification of traditional Jacobi-quartic coordinates was performed by Duquesne [10] followed by an improvement in [4]. A fast doubling formula was proposed by the authors [14]. This formula was adapted to the corresponding coordinates for the study by Bernstein and Lange [4]. For the completeness of this paper we recall the modified Jacobi-quartic coordinates (with some minor differences). Two points;

$$(X_1\colon Y_1\colon Z_1\colon U_1\colon V_1\colon W_1), \quad (X_2\colon Y_2\colon Z_2\colon U_2\colon V_2\colon W_2)$$

with

$$U_1 \leftarrow X_1^2, \quad V_1 \leftarrow Z_1^2, \quad W_1 \leftarrow 2X_1Z_1$$
$$U_2 \leftarrow X_2^2, \quad V_2 \leftarrow Z_2^2, \quad W_2 \leftarrow 2X_2Z_2$$

can be added as follows. The inputs that set $Z_3 = 0$ should be handled separately.

$$A \leftarrow W_1W_2, \quad B \leftarrow Y_1Y_2, \quad C \leftarrow V_1V_2, \quad D \leftarrow U_1U_2,$$

$$X_3 \leftarrow (W_1 + Y_1)(W_2 + Y_2) - A - B, \quad Z_3 \leftarrow 2(C - D),$$
$$Y_3 \leftarrow 2((C + D)(2B + aA) + A((U_1 + V_1)(U_2 + V_2) - (C + D))),$$
$$U_3 \leftarrow X_3^2, \quad V_3 \leftarrow Z_3^2, \quad W_3 \leftarrow (X_3 + Z_3)^2 - U_3 - V_3.$$

The (unified) addition on modified Jacobi-quartic coordinates costs 8M+3S+1D as reported in [10, 4]. For a 7M+3S+1D mixed addition see [4, 14]. A point;

$$(X_1\colon Y_1\colon Z_1\colon U_1\colon V_1\colon W_1)$$

can be added to itself as follows. (The derivation of this formula is given in Appendix-C).

$$A \leftarrow U_1 + V_1, \quad X_3 \leftarrow Y_1W_1, \quad Z_3 \leftarrow A(V_1 - U_1), \quad U_3 \leftarrow X_3^2,$$

$$V_3 \leftarrow Z_3^2, \quad B \leftarrow U_3 + V_3, \quad W_3 \leftarrow (X_3 + Z_3)^2 - B, \quad Y_3 \leftarrow 2(Y_1A)^2 - B.$$

The operation count shows that Jacobi-quartic doubling on modified coordinates costs **3M+4S**. One can reduce the number of coordinates without storing $W$. This does not effect doubling, readdition and mixed addition costs. However, the addition costs 8M+4S+1D in this case.

# 5 Conclusion

We improved the latest operation counts for Hessian and Jacobi-intersection forms by modifying the traditional coordinates. We derived a unified addition formula for Hessian form and showed that this addition is faster than the unified addition in Weierstrass form. The speed comparison of point addition formulae for different forms are given in Figure 1. When compared to the other forms, 6M+6S modified Hessian addition can be the fastest in some cases though it uses 6 extra variables for storing each point. For instance, in Figure 1 we see;

- If S/M=0.67, modified Hessian addition is faster than all other additions except the inverted Edwards addition. Modified Hessian addition becomes the fastest whenever D/M>0.33.
- If S/M=0.8, modified Hessian addition is faster than Edwards, modified Jacobi-intersection and Weierstrass addition for all D/M values. It is faster than modified Jacobi-quartic addition when D/M>0.4. It yields the same complexity with inverted Edwards addition when D/M=1.

If S/M=1, the modified coordinates for Hessian form do not provide any advantages over the traditional coordinates and it will be slower than Edwards, inverted Edwards and Jacobi-quartic additions for all D/M values. On the other hand, a field squaring can be performed faster than a field multiplication in most environments. For instance a low S/M is reported in [16]. However, the S/M value may vary depending on the hardware environment and the type of field reduction modulus used.

It is known that in memory limited environments (such as smartcards), there is not enough space for storing large precomputation tables. For these environments, point multiplication with non-adjacent form without precomputation technique is a convenient selection. This algorithm requires 1 doubling and 1/3 mixed addition per key bit. The operation counts for the various systems with respect to this algorithm are depicted in Table 1."DBL" stands for doubling. "mADD" stands for mixed addition. The ratios for each column such as (.5, .67) are the D/M, S/M values respectively. The rows are sorted with respect to the column (0, .8) in descending order. Some forms have alternative versions due to alternative doubling and/or addition formulae for different S/M and D/M values. The main outcomes are as follows.

- Modified Jacobi-quartic coordinates (which uses our fast doubling formula) provides the fastest timings for
  - D/M>0.1 when S/M=0.67,
  - D/M>0.2 when S/M=0.8,
  - D/M>0.33 when S/M=1.

Bernstein and Lange [12, 13] provide a detailed cost analysis for "signed 4-bit sliding windows" point multiplication algorithm (for 256-bit scalars). The

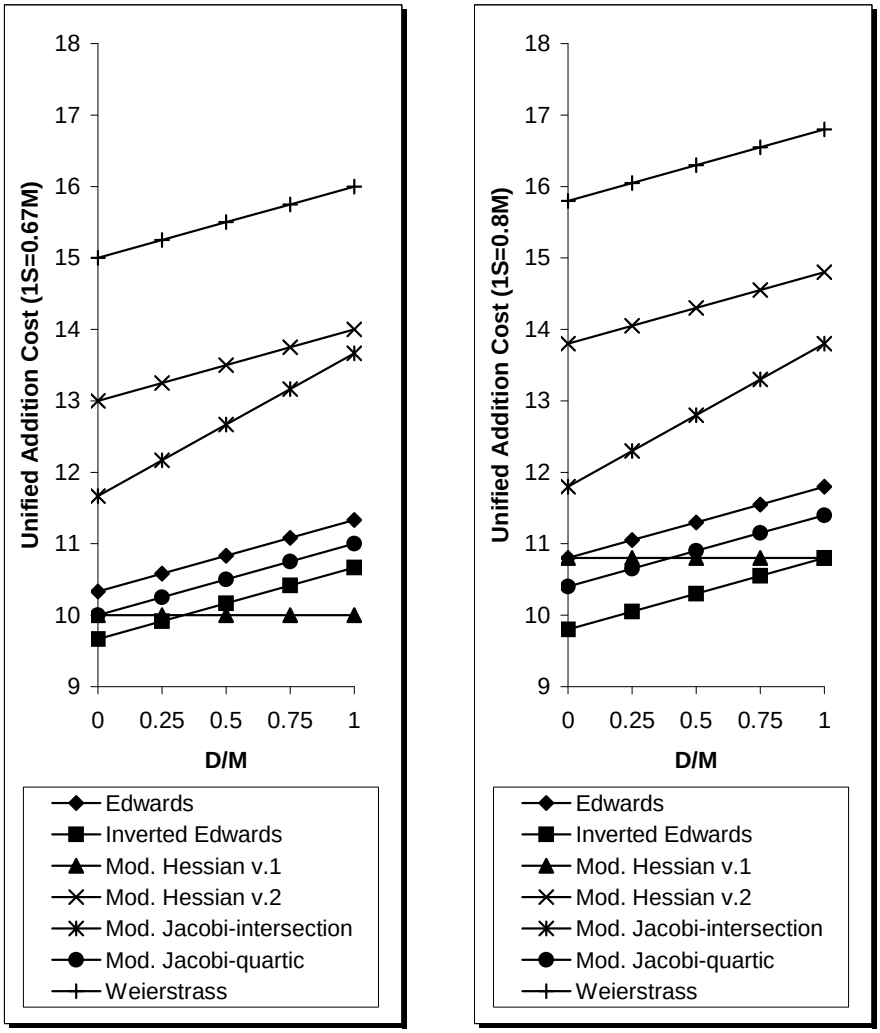


Figure 1: Performance comparison of unified additions for various forms with different S/M, D/M values. The left figure contains the ratios for S/M=0.67 and all D/M values. The operation counts are as follows: Edwards 7M+5S+1D, modified Hessian 6M+6S (v.1) or 9M+6S+1D (v.2), inverted Edwards 9M+1S+1D (from [4, 5, 17]), modified Jacobi-intersection 11M+1S+2D, modified Jacobi-quartic 8M+3S+1D (from [10, 4]), Weierstrass 11M+6S+1D (from [8, 4]). The right figure contains the ratios for S/M=0.80 and all D/M values. The operation counts are the same except the Edwards addition which costs 10M+1S+1D (from [13, 4, 12]) in this case.

algorithm requires 0.98 doublings, 0.17 readditions, 0.025 mixed additions and 0.0035 additions per key bit. We use their analysis to report current ranking between different coordinate systems. The operation counts for the various systems with respect to this algorithm are depicted in Table 2."DBL" stands for doubling. "mADD" stands for mixed addition."reADD" stands for readdition. "ADD" stands for addition. The underlined values are the fastest timings. The rows are sorted with respect to the column (0, .8) in descending order. The main outcomes are as follows.

- Modified Jacobi-quartic coordinates (which uses our fast doubling formula) provides the fastest timings for
  - D/M>0.07 when S/M=0.67,
  - D/M>0.12 when S/M=0.8,
  - D/M>0.2 when S/M=1.
- If S/M$\leq$0.8 and D/M$\approx$0, performance of the modified Jacobi-intersection coordinates is very close to the other fastest systems.

To sum up, we provided new coordinate systems for Hessian and Jacobi-intersection forms. We provided a detailed performance evaluation of various systems for elliptic curve point multiplication algorithms. We pointed to the S/M and D/M scenarios where modified Hessian (unified) addition is the fastest. We also determined when modified Jacobi-quartic form (which uses our doubling formulae in [14]) provides the fastest timings for speed oriented point multiplications algorithms.

| System | DBL | | | mADD | | | NON-ADJACENT FORM WITHOUT PRECOMPUTATION (1 doublings, 0.33 mixed-additions) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | M | S | D | M | S | D | M | S | D | 1, 1 | 1, .8 | 1, .66 | .5, 1 | .5, .8 | .5, .66 | 0, 1 | 0, .8 | 0, .66 |
| **Projective** | 5 | 6 | 1 | 9 | 2 | 0 | 8.000 | 6.667 | 1.000 | 15.667 | 14.333 | 13.400 | 15.167 | 13.833 | 12.900 | 14.667 | 13.333 | 12.400 |
| **Projective (a=-3)** | 7 | 3 | 0 | 9 | 2 | 0 | 10.000 | 3.667 | 0.000 | 13.667 | 12.933 | 12.420 | 13.667 | 12.933 | 12.420 | 13.667 | 12.933 | 12.420 |
| **Doche/Icart/Kohel-3** | 2 | 7 | 2 | 10 | 6 | 1 | 5.333 | 9.000 | 2.333 | 16.667 | 14.867 | 13.607 | 15.500 | 13.700 | 12.440 | 14.333 | 12.533 | 11.273 |
| **Jacobi-quartic v.2** | 1 | 9 | 0 | 8 | 3 | 1 | 3.667 | 10.000 | 0.333 | 14.000 | 12.000 | 10.600 | 13.833 | 11.833 | 10.433 | 13.667 | 11.667 | 10.267 |
| **Hessian v.1** | 7 | 1 | 0 | 10 | 0 | 0 | 10.333 | 1.000 | 0.000 | 11.333 | 11.133 | 10.993 | 11.333 | 11.133 | 10.993 | 11.333 | 11.133 | 10.993 |
| **Hessian v.3** | 3 | 6 | 0 | 10 | 0 | 0 | 6.333 | 6.000 | 0.000 | 12.333 | 11.133 | 10.293 | 12.333 | 11.133 | 10.293 | 12.333 | 11.133 | 10.293 |
| **Hessian v.2** | 7 | 1 | 0 | 5 | 6 | 0 | 8.667 | 3.000 | 0.000 | 11.667 | 11.067 | 10.647 | 11.667 | 11.067 | 10.647 | 11.667 | 11.067 | 10.647 |
| **Hessian v.4** | 3 | 6 | 0 | 5 | 6 | 0 | 4.667 | 8.000 | 0.000 | 12.667 | 11.067 | 9.947 | 12.667 | 11.067 | 9.947 | 12.667 | 11.067 | 9.947 |
| **Modified Hessian** | 3 | 6 | 0 | 5 | 6 | 0 | 4.667 | 8.000 | 0.000 | 12.667 | 11.067 | 9.947 | 12.667 | 11.067 | 9.947 | 12.667 | 11.067 | 9.947 |
| **Jacobian** | 1 | 8 | 1 | 7 | 4 | 0 | 3.333 | 9.333 | 1.000 | 13.667 | 11.800 | 10.493 | 13.167 | 11.300 | 9.993 | 12.667 | 10.800 | 9.493 |
| **Jacobian (a=-3)** | 3 | 5 | 0 | 7 | 4 | 0 | 5.333 | 6.333 | 0.000 | 11.667 | 10.400 | 9.513 | 11.667 | 10.400 | 9.513 | 11.667 | 10.400 | 9.513 |
| **Jacobi-quartic v.1** | 1 | 7 | 2 | 8 | 3 | 1 | 3.667 | 8.000 | 2.333 | 14.000 | 12.400 | 11.280 | 12.833 | 11.233 | 10.113 | 11.667 | 10.067 | 8.947 |
| **Jacobi-intersection v.1** | 3 | 4 | 0 | 10 | 2 | 1 | 6.333 | 4.667 | 0.333 | 11.333 | 10.400 | 9.747 | 11.167 | 10.233 | 9.580 | 11.000 | 10.067 | 9.413 |
| **Jacobi-intersection v.2** | 2 | 5 | 1 | 10 | 2 | 1 | 5.333 | 5.667 | 1.333 | 12.333 | 11.200 | 10.407 | 11.667 | 10.533 | 9.740 | 11.000 | 9.867 | 9.073 |
| **Doche/Icart/Kohel-2** | 2 | 5 | 2 | 8 | 4 | 1 | 4.667 | 6.333 | 2.333 | 13.333 | 12.067 | 11.180 | 12.167 | 10.900 | 10.013 | 11.000 | 9.733 | 8.847 |
| **Modified Jacobi-intersection** | 2 | 5 | 1 | 10 | 1 | 2 | 5.333 | 5.333 | 1.667 | 12.333 | 11.267 | 10.520 | 11.500 | 10.433 | 9.687 | 10.667 | 9.600 | 8.853 |
| **Edwards v.2** | 3 | 4 | 0 | 6 | 5 | 1 | 5.000 | 5.667 | 0.333 | 11.000 | 9.867 | 9.073 | 10.833 | 9.700 | 8.907 | 10.667 | 9.533 | 8.740 |
| **Edwards v.1** | 3 | 4 | 0 | 9 | 1 | 1 | 6.000 | 4.333 | 0.333 | 10.667 | 9.800 | 9.193 | 10.500 | 9.633 | 9.027 | 10.333 | 9.467 | 8.860 |
| **Modified Jacobi-quartic** | 3 | 4 | 0 | 7 | 3 | 1 | 5.333 | 5.000 | 0.333 | 10.667 | 9.667 | 8.967 | 10.500 | 9.500 | 8.800 | 10.333 | 9.333 | 8.633 |
| **Inverted-Edwards** | 3 | 4 | 1 | 8 | 1 | 1 | 5.667 | 4.333 | 1.333 | 11.333 | 10.467 | 9.860 | 10.667 | 9.800 | 9.193 | 10.000 | 9.133 | 8.527 |

Table 1: Point multiplication costs per key bit for "Non-adjacent Form without Precomputation" method.

| System | DBL | | | reADD | | | mADD | | | ADD | | | SIGNED 4-BIT SLIDING WINDOWS (0.98 doublings, 0.17 re-additions, 0.025 mixed-additons, 0.0035 additions) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | M | S | D | M | S | D | M | S | D | M | S | D | M | S | D | 1, 1 | 1, .8 | 1, .67 | .5, 1 | .5, .8 | .5, .67 | 0, 1 | 0, .8 | 0, .67 |
| **Projective** | 5 | 6 | 1 | 12 | 2 | 0 | 9 | 2 | 0 | 12 | 2 | 0 | 7.168 | 6.283 | 0.982 | 14.433 | 13.177 | 12.339 | 13.942 | 12.685 | 11.848 | 13.451 | 12.194 | 11.357 |
| **Projective (a=-3)** | 7 | 3 | 0 | 12 | 2 | 0 | 9 | 2 | 0 | 12 | 2 | 0 | 9.133 | 3.336 | 0.000 | 12.468 | 11.801 | 11.357 | 12.468 | 11.801 | 11.357 | 12.468 | 11.801 | 11.357 |
| **Jacobi-quartic v.2** | 1 | 9 | 0 | 8 | 3 | 1 | 8 | 3 | 1 | 10 | 3 | 1 | 2.543 | 9.424 | 0.194 | 12.161 | 10.276 | 9.020 | 12.064 | 10.179 | 8.922 | 11.967 | 10.082 | 8.825 |
| **Hessian v.3** | 3 | 6 | 0 | 12 | 0 | 0 | 10 | 0 | 0 | 12 | 0 | 0 | 5.228 | 5.895 | 0.000 | 11.122 | 9.943 | 9.157 | 11.122 | 9.943 | 9.157 | 11.122 | 9.943 | 9.157 |
| **Hessian v.1** | 7 | 1 | 0 | 12 | 0 | 0 | 10 | 0 | 0 | 12 | 0 | 0 | 9.157 | 0.982 | 0.000 | 10.140 | 9.943 | 9.812 | 10.140 | 9.943 | 9.812 | 10.140 | 9.943 | 9.812 |
| **Hessian v.2** | 7 | 1 | 0 | 12 | 0 | 0 | 5 | 6 | 0 | 12 | 0 | 0 | 9.034 | 1.130 | 0.000 | 10.164 | 9.938 | 9.788 | 10.164 | 9.938 | 9.788 | 10.164 | 9.938 | 9.788 |
| **Hessian v.4** | 3 | 6 | 0 | 12 | 0 | 0 | 5 | 6 | 0 | 12 | 0 | 0 | 5.105 | 6.042 | 0.000 | 11.147 | 9.938 | 9.133 | 11.147 | 9.938 | 9.133 | 11.147 | 9.938 | 9.133 |
| **Jacobian** | 1 | 8 | 1 | 10 | 4 | 0 | 7 | 4 | 0 | 11 | 5 | 0 | 2.854 | 8.639 | 0.982 | 12.475 | 10.748 | 9.596 | 11.984 | 10.256 | 9.104 | 11.493 | 9.765 | 8.613 |
| **Modified Hessian** | 3 | 6 | 0 | 6 | 6 | 0 | 5 | 6 | 0 | 6 | 6 | 0 | 4.088 | 7.059 | 0.000 | 11.147 | 9.735 | 8.794 | 11.147 | 9.735 | 8.794 | 11.147 | 9.735 | 8.794 |
| **Doche/Icart/Kohel-3** | 2 | 7 | 2 | 7 | 4 | 1 | 10 | 6 | 1 | 11 | 6 | 1 | 3.412 | 7.710 | 2.159 | 13.280 | 11.739 | 10.711 | 12.201 | 10.659 | 9.631 | 11.121 | 9.580 | 8.552 |
| **Jacobian (a=-3)** | 3 | 5 | 0 | 10 | 4 | 0 | 7 | 4 | 0 | 11 | 5 | 0 | 4.818 | 5.692 | 0.000 | 10.511 | 9.372 | 8.613 | 10.511 | 9.372 | 8.613 | 10.511 | 9.372 | 8.613 |
| **Doche/Icart/Kohel-2** | 2 | 5 | 2 | 12 | 5 | 1 | 8 | 4 | 1 | 12 | 5 | 1 | 4.196 | 5.858 | 2.159 | 12.213 | 11.042 | 10.261 | 11.134 | 9.962 | 9.181 | 10.054 | 8.883 | 8.102 |
| **Jacobi-intersection v.1** | 3 | 4 | 0 | 11 | 2 | 1 | 10 | 2 | 1 | 13 | 2 | 1 | 5.065 | 4.318 | 0.194 | 9.577 | 8.714 | 8.138 | 9.480 | 8.617 | 8.041 | 9.383 | 8.520 | 7.944 |
| **Jacobi-quartic v.1** | 1 | 7 | 2 | 8 | 3 | 1 | 8 | 3 | 1 | 10 | 3 | 1 | 2.543 | 7.459 | 2.159 | 12.161 | 10.669 | 9.674 | 11.081 | 9.590 | 8.595 | 10.002 | 8.510 | 7.515 |
| **Jacobi-intersection v.2** | 2 | 5 | 1 | 11 | 2 | 1 | 10 | 2 | 1 | 13 | 2 | 1 | 4.083 | 5.300 | 1.177 | 10.560 | 9.500 | 8.793 | 9.971 | 8.911 | 8.205 | 9.383 | 8.323 | 7.616 |
| **Edwards v.2** | 3 | 4 | 0 | 7 | 5 | 1 | 6 | 5 | 1 | 7 | 5 | 1 | 4.282 | 4.900 | 0.194 | 9.376 | 8.396 | 7.743 | 9.279 | 8.299 | 7.646 | 9.182 | 8.202 | 7.549 |
| **Edwards v.1** | 3 | 4 | 0 | 10 | 1 | 1 | 9 | 1 | 1 | 10 | 1 | 1 | 4.864 | 4.124 | 0.194 | 9.182 | 8.357 | 7.807 | 9.085 | 8.260 | 7.710 | 8.988 | 8.163 | 7.613 |
| **Modified Jacobi-intersection** | 2 | 5 | 1 | 11 | 1 | 2 | 10 | 1 | 2 | 11 | 1 | 2 | 4.076 | 5.106 | 1.371 | 10.553 | 9.531 | 8.851 | 9.867 | 8.846 | 8.165 | 9.182 | 8.161 | 7.480 |
| **Modified Jacobi-quartic** | 3 | 4 | 0 | 8 | 3 | 1 | 7 | 3 | 1 | 8 | 3 | 1 | 4.476 | 4.512 | 0.194 | 9.182 | 8.280 | 7.678 | 9.085 | 8.183 | 7.581 | 8.988 | 8.085 | 7.484 |
| **Inverted-Edwards** | 3 | 4 | 1 | 9 | 1 | 1 | 8 | 1 | 1 | 9 | 1 | 1 | 4.670 | 4.124 | 1.177 | 9.970 | 9.146 | 8.596 | 9.382 | 8.557 | 8.007 | 8.794 | 7.969 | 7.419 |

Table 2: Point multiplication costs per key bit for "Signed 4-bit Sliding Windows" method.

# References


[1] Chudnovsky, D.V., Chudnovsky, G.V.: Sequences of numbers generated by addition in formal groups and new primality and factorization tests. Adv. Appl. Math. **7**(4) (1986) 385–434

[2] Cohen, H., Miyaji, A., Ono, T.: Efficient elliptic curve exponentiation using mixed coordinates. In: ASIACRYPT, London, UK, Springer-Verlag (1998) 51–65

[3] Doche, C., Icart, T., Kohel, D.R.: Efficient scalar multiplication by isogeny decompositions. In: Public Key Cryptography. Volume 3958 of Lecture Notes in Computer Science., Springer (2006) 191–206

[4] Bernstein, D.J., Lange, T.: Explicit-formulas database, *Accessible through:* http://hyperelliptic.org/EFD (2007)

[5] Bernstein, D.J., Birkner, P., Lange, T., Peters, C.: Optimizing double-base elliptic-curve single-scalar multiplication. In: INDOCRYPT. Volume This volume of Lecture Notes in Computer Science., Springer (2007)

[6] Joye, M., Quisquater, J.J.: Hessian elliptic curves and side-channel attacks. In: CHES. Volume 2162 of Lecture Notes in Computer Science., Springer (2001) 402–410

[7] Liardet, P.Y., Smart, N.P.: Preventing SPA/DPA in ECC systems using the Jacobi form. In: CHES. Volume 2162 of Lecture Notes in Computer Science., Springer (2001) 391–401

[8] Brier, E., Joye, M.: Weierstrass elliptic curves and side-channel attacks. In: PKC, London, UK, Springer-Verlag (2002) 335–345

[9] Billet, O., Joye, M.: The Jacobi model of an elliptic curve and side-channel analysis. In: AAECC. Volume 2643 of Lecture Notes in Computer Science., Springer (2003) 34–42

[10] Duquesne, S.: Improving the arithmetic of elliptic curves in the Jacobi model. Inf. Process. Lett. **104**(3) (2007) 101–105

[11] Edwards, H.M.: A normal form for elliptic curves. Bulletin of the AMS **44**(3) (2007) 393–422

[12] Bernstein, D.J., Lange, T.: Faster addition and doubling on elliptic curves. Cryptology ePrint Archive, Report 2007/286 (2007) `http://eprint.iacr.org/`.

[13] Bernstein, D.J., Lange, T.: Faster addition and doubling on elliptic curves. In: ASIACRYPT. Volume 4833 of Lecture Notes in Computer Science., Berlin Heidelberg, Springer-Verlag (2007) 29–50

[14] Hisil, H., Carter, G., Dawson, E.: New formulae for efficient elliptic curve arithmetic. In: INDOCRYPT. Volume 4859 of Lecture Notes in Computer Science., Springer (2007) 138–151

[15] Smart, N.P.: The Hessian form of an elliptic curve. In: CHES. Volume 2162 of Lecture Notes in Computer Science., Springer (2001) 118–125

[16] Bernstein, D.J.: Curve25519: New Diffie-Hellman speed records. In Yung, M., Dodis, Y., Kiayias, A., Malkin, T., eds.: Public Key Cryptography. Volume 3958 of Lecture Notes in Computer Science., Springer (2006) 207–228

[17] Bernstein, D.J., Lange, T.: Inverted edwards coordinates. Cryptology ePrint Archive, Report 2007/410 (2007) `http://eprint.iacr.org/`.

# APPENDIX-A

We give a step by step derivation of 9M+6S+1D unified addition formula for Hessian form. The unified addition formula can be derived from the "chord-and-tangent" rule however it is easier to explain the procedure if we use the Cauchy-Sylvester formula which is as follows.

$$\begin{aligned} X_3 &= Y_1^2X_2Z_2 - X_1Z_1Y_2^2 \\ Y_3 &= X_1^2Y_2Z_2 - Y_1Z_1X_2^2 \\ Z_3 &= Z_1^2X_2Y_2 - X_1Y_1Z_2^2 \end{aligned}$$

Note, it is enough to work with the coordinates $X_3$ and $Z_3$. It is easy to derive $Y_3$ via the symmetry involved in the curve equation $x^3 + y^3 + 1 = 3dxy$.

**Step1:** Multiply each coordinate with $(X_1X_2)^3 - (Y_1Y_2)^3$ assuming $(X_1X_2)^3 \neq (Y_1Y_2)^3$.

$$\begin{aligned} X_3 &= ((X_1X_2)^3 - (Y_1Y_2)^3)(Y_1^2X_2Z_2 - X_1Z_1Y_2^2) \\ Z_3 &= ((X_1X_2)^3 - (Y_1Y_2)^3)(Z_1^2X_2Y_2 - X_1Y_1Z_2^2) \end{aligned}$$

**Step2:** The quantity $(3dX_1X_2Y_1^3Y_2^3Z_1Z_2 - 3dX_1X_2Y_1^3Y_2^3Z_1Z_2) + (X_1^3X_2Y_1^2Y_2^3Z_2 - X_1^3X_2Y_1^2Y_2^3Z_2) + (X_1X_2^3Y_1^3Y_2^2Z_1 - X_1X_2^3Y_1^3Y_2^2Z_1)$ is obviously zero. Add it to $X_3$.

$$\begin{aligned} X_3 &= ((X_1X_2)^3 - (Y_1Y_2)^3)(Y_1^2X_2Z_2 - X_1Z_1Y_2^2) + \\ &\quad 3dX_1X_2Y_1^3Y_2^3Z_1Z_2 - 3dX_1X_2Y_1^3Y_2^3Z_1Z_2 + \\ &\quad X_1^3X_2Y_1^2Y_2^3Z_2 - X_1^3X_2Y_1^2Y_2^3Z_2 + \\ &\quad X_1X_2^3Y_1^3Y_2^2Z_1 - X_1X_2^3Y_1^3Y_2^2Z_1 \\ Z_3 &= ((X_1X_2)^3 - (Y_1Y_2)^3)(Z_1^2X_2Y_2 - X_1Y_1Z_2^2) \end{aligned}$$

**Step3:** Rearrange the terms in $X_3$ as follows.

$$\begin{aligned} X_3 &= X_1X_2^3Y_2^2Z_1(3dX_1Y_1Z_1 - X_1^3 - Y_1^3) + \\ &\quad X_2Y_1^2Y_2^3Z_2(3dX_1Y_1Z_1 - X_1^3 - Y_1^3) - \\ &\quad X_1Y_1^3Y_2^2Z_1(3dX_2Y_2Z_2 - X_2^3 - Y_2^3) - \\ &\quad X_1^3X_2Y_1^2Z_2(3dX_2Y_2Z_2 - X_2^3 - Y_2^3) + \\ &\quad 3dX_1^3X_2^2Y_1^2Y_2Z_2^2 - 3dX_1^2X_2^3Y_1Y_2^2Z_1^2 \\ Z_3 &= ((X_1X_2)^3 - (Y_1Y_2)^3)(Z_1^2X_2Y_2 - X_1Y_1Z_2^2) \end{aligned}$$

**Step4:** Using the projective curve equation replace $(3dX_1Y_1Z_1 - X_1^3 - Y_1^3)$ by $Z_1^3$ and replace $(3dX_2Y_2Z_2 - X_2^3 - Y_2^3)$ by $Z_2^3$.

$$\begin{aligned} X_3 &= X_1X_2^3Y_2^2Z_1^4 + X_2Y_1^2Y_2^3Z_1^3Z_2 - 3dX_1^2X_2^3Y_1Y_2^2Z_1^2 - \\ &\quad X_1Y_1^3Y_2^2Z_1Z_2^3 - X_1^3X_2Y_1^2Z_2^4 + 3dX_1^3X_2^2Y_1^2Y_2Z_2^2 \\ Z_3 &= ((X_1X_2)^3 - (Y_1Y_2)^3)(Z_1^2X_2Y_2 - X_1Y_1Z_2^2) \end{aligned}$$

**Step5:** Factorize $X_3$ over the base field.

$$\begin{aligned} X_3 &= (X_1X_2^2Y_2Z_1^2 + X_1^2X_2Y_1Z_2^2 + Y_1^2Y_2^2Z_1Z_2 - 3dX_1^2X_2^2Y_1Y_2) \cdot \\ &\quad (Z_1^2X_2Y_2 - X_1Y_1Z_2^2) \\ Z_3 &= ((X_1X_2)^3 - (Y_1Y_2)^3)(Z_1^2X_2Y_2 - X_1Y_1Z_2^2) \end{aligned}$$

**Step6:** Divide each coordinate to $(Z_1^2X_2Y_2 - X_1Y_1Z_2^2)$ which is assumed to be nonzero.

$$\begin{aligned} X_3 &= X_1X_2^2Y_2Z_1^2 + X_1^2X_2Y_1Z_2^2 + Y_1^2Y_2^2Z_1Z_2 - 3dX_1^2X_2^2Y_1Y_2 \\ Z_3 &= (X_1X_2)^3 - (Y_1Y_2)^3 \end{aligned}$$

**Step7:** Generate $Y_3$ from the symmetry.

$$\begin{aligned} X_3 &= X_1X_2^2Y_2Z_1^2 + X_1^2X_2Y_1Z_2^2 + Y_1^2Y_2^2Z_1Z_2 - 3dX_1^2X_2^2Y_1Y_2 \\ Y_3 &\quad -Y_1Y_2^2X_2Z_1^2 - X_1Z_2^2Y_1^2Y_2 - X_1^2X_2^2Z_1Z_2 + 3dX_1X_2Y_1^2Y_2^2 \\ Z_3 &= (X_1X_2)^3 - (Y_1Y_2)^3 \end{aligned}$$

**Step8:** Rearrange terms in $X_3$ and $Y_3$. This gives the unified addition formula for Hessian form.

$$\begin{aligned} X_3 &= X_1X_2(X_1Y_1Z_2^2 + X_2Y_2Z_1^2 - 3dX_1Y_1X_2Y_2) + (Y_1Y_2)^2Z_1Z_2 \\ Y_3 &= -Y_1Y_2(X_1Y_1Z_2^2 + X_2Y_2Z_1^2 - 3dX_1Y_1X_2Y_2) - (X_1X_2)^2Z_1Z_2 \\ Z_3 &= (X_1X_2)^3 - (Y_1Y_2)^3 \end{aligned}$$

We explain the derivation in projective model. Of course, an analogous derivation can be done in the affine model. In that case, one will find the

following affine formula for Hessian addition.

$$\begin{aligned} x_3 &= \frac{-x_1^2x_2^2 - x_1y_1^2y_2 - x_2y_1y_2^2 + 3dx_1x_2y_1^2y_2^2}{x_1^3x_2^3 - y_1^3y_2^3} \\ y_3 &= \frac{x_1^2x_2y_1 + x_1x_2^2y_2 - 3dx_1^2x_2^2y_1y_2 + y_1^2y_2^2}{x_1^3x_2^3 - y_1^3y_2^3} \end{aligned}$$

# APPENDIX-B

We give a step by step derivation of 2M+5S+1D doubling formula for Jacobi-quartic form. The original formula, described by Liardet and Smart [7], is as follows.

$$\begin{aligned} S_3 &= 2S_1C_1D_1T_1 \\ C_3 &= -S_1^2D_1^2 + C_1^2T_1^2 \\ D_3 &= S_1^2D_1^2 - C_1^2T_1^2 + 2C_1^2D_1^2 \\ T_3 &= S_1^2D_1^2 + C_1^2T_1^2 \end{aligned}$$

**Step1:** Using the defining equations $S_1^2 + C_1^2 = T_1^2$ and $aS_1^2 + D_1^2 = T_1^2$, replace all instances of $S_1^2D_1^2$ with $(T_1^2 - C_1^2)D_1^2$ and $C_1^2T_1^2$ with $C_1^2(aS_1^2 + D_1^2)$.

$$\begin{aligned} S_3 &= 2S_1C_1D_1T_1 \\ C_3 &= -(T_1^2 - C_1^2)D_1^2 + C_1^2(aS_1^2 + D_1^2) \\ D_3 &= (T_1^2 - C_1^2)D_1^2 - C_1^2(aS_1^2 + D_1^2) + 2C_1^2D_1^2 \\ T_3 &= (T_1^2 - C_1^2)D_1^2 + C_1^2(aS_1^2 + D_1^2) \end{aligned}$$

**Step2:** Replace the term $2C_1^2D_1^2$ with $2C_1^2(S_1^2 + C_1^2 - aS_1^2)$.

$$\begin{aligned} S_3 &= 2S_1C_1D_1T_1 \\ C_3 &= -D_1^2T_1^2 + aS_1^2C_1^2 + 2C_1^2(S_1^2 + C_1^2 - aS_1^2) \\ D_3 &= D_1^2T_1^2 - aS_1^2C_1^2 \\ T_3 &= D_1^2T_1^2 + aS_1^2C_1^2 \end{aligned}$$

**Step3:** Rearrange terms. This gives the desired doubling formula.

$$\begin{aligned} S_3 &= 2S_1C_1D_1T_1 \\ C_3 &= -D_1^2T_1^2 - aS_1^2C_1^2 + 2(S_1^2C_1^2 + C_1^4) \\ D_3 &= D_1^2T_1^2 - aS_1^2C_1^2 \\ T_3 &= D_1^2T_1^2 + aS_1^2C_1^2 \end{aligned}$$

The corresponding affine formula for Jacobi-intersection doubling is as follows.

$$s_3 = \frac{2s_1c_1d_1}{d_1^2 + as_1^2c_1^2}$$

$$
\begin{aligned}
c_3 &= \frac{2c_1^4 + 2s_1^2c_1^2 - d_1^2 - as_1^2c_1^2}{d_1^2 + as_1^2c_1^2} \\
d_3 &= \frac{d_1^2 - as_1^2c_1^2}{d_1^2 + as_1^2c_1^2}
\end{aligned}
$$

# APPENDIX-C

We give a step by step derivation of 3M+4S doubling formula for Jacobi-quartic form. The original formula, described by Billet and Joye [9], is as follows.

$$
\begin{aligned}
X_3 &= 2X_1Y_1Z_1 \\
Y_3 &= 2aX_1^2Z_1^6 + 4X_1^4Z_1^4 + Y_1^2Z_1^4 + 2aX_1^6Z_1^2 + X_1^4Y_1^2 \\
Z_3 &= Z_1^4 - X_1^4
\end{aligned}
$$

**Step1:** Modify the point $(X_3 : Y_3 : Z_3)$ to $(-X_3 : Y_3 : -Z_3)$. These two points correspond to the same affine point.

$$
\begin{aligned}
X_3 &= -2X_1Y_1Z_1 \\
Y_3 &= 2aX_1^2Z_1^6 + 4X_1^4Z_1^4 + Y_1^2Z_1^4 + 2aX_1^6Z_1^2 + X_1^4Y_1^2 \\
Z_3 &= X_1^4 - Z_1^4
\end{aligned}
$$

**Step2:** Organize $X_3$ and $Z_3$. Here, $Y_3$ should be computed after $X_3$ and $Z_3$.

$$
\begin{aligned}
X_3 &= Y_1((X_1^2 + Z_1^2)) - Y_1(X_1 + Z_1)^2 \\
Z_3 &= (X_1^2 + Z_1^2)(X_1^2 - Z_1^2) \\
Y_3 &= 2aX_1^2Z_1^6 + 4X_1^4Z_1^4 + Y_1^2Z_1^4 + 2aX_1^6Z_1^2 + X_1^4Y_1^2
\end{aligned}
$$

**Step3:** Use the curve equation, $Y_1^2 = X_1^4 + 2aX_1^2Z_1^2 + Z_1^4$, to find a suitable polynomial representation for $Y_3$.

$$
\begin{aligned}
X_3 &= Y_1((X_1^2 + Z_1^2)) - Y_1(X_1 + Z_1)^2 \\
\\
Z_3 &= (X_1^2 + Z_1^2)(X_1^2 - Z_1^2) \\
\\
Y_3 &= 2aX_1^2Z_1^6 + 4X_1^4Z_1^4 + Y_1^2Z_1^4 + 2aX_1^6Z_1^2 + X_1^4Y_1^2 \\
&= 2aX_1^2Z_1^6 + 4X_1^4Z_1^4 + 2aX_1^6Z_1^2 + (X_1^4 + Z_1^4)Y_1^2 \\
&\equiv 2aX_1^2Z_1^6 + 4X_1^4Z_1^4 + 2aX_1^6Z_1^2 + (X_1^4 + Z_1^4)(Z_1^4 + 2aX_1^2Z_1^2 + X_1^4) \\
&\equiv Z_1^8 + 4aX_1^2Z_1^6 + 6X_1^4Z_1^4 + 4aX_1^6Z_1^2 + X_1^8 \\
&\equiv 2(Y_1Z_1^2 + X_1^2Y_1)^2 - (Z_1^8 - 2X_1^4Z_1^4 + X_1^8) - (4X_1^2Y_1^2Z_1^2) \\
&\equiv 2(Y_1(X_1^2 + Z_1^2))^2 - X_3^2 - Z_3^2
\end{aligned}
$$

The corresponding affine formula for Jacobi-quartic doubling is as follows.

$$\begin{aligned} x_3 &= \frac{-2x_1y_1}{x_1^4-1} \\ y_3 &= \frac{2(y_1(x_1^2+1))^2-(-2x_1y_1)^2-(x_1^4-1)^2}{(x_1^4-1)^2} \end{aligned}$$