Faster Group Operations on Elliptic Curves

Huseyin Hisil[1] **Kenneth Koon-Ho Wong**[1] **Gary Carter**[1] **Ed Dawson**[1]

[1] Information Security Institute,
Queensland University of Technology,
Brisbane, QLD, Australia, 4000
{h.hisil, kk.wong, g.carter, e.dawson}@qut.edu.au

Abstract

This paper improves implementation techniques of Elliptic Curve Cryptography. We introduce new formulae and algorithms for the group law on Jacobi quartic, Jacobi intersection, Edwards, and Hessian curves. The proposed formulae and algorithms can save time in suitable point representations. To support our claims, a cost comparison is made with classic scalar multiplication algorithms using previous and current operation counts. Most notably, the best speeds are obtained from Jacobi quartic curves which provide the fastest timings for most scalar multiplication strategies benefiting from the proposed[1] $2\mathbf{M} + 5\mathbf{S} + 1\mathbf{D}$ point doubling and $7\mathbf{M} + 3\mathbf{S} + 1\mathbf{D}$ point addition algorithms. Furthermore, the new addition algorithm provides an efficient way to protect against side channel attacks which are based on simple power analysis (SPA).

Keywords: Efficient elliptic curve arithmetic, unified addition, side channel attack.

1 Introduction

From the advent of elliptic curve cryptosystems, independently by Miller (1986) and Koblitz (1987) to date, the arithmetic of elliptic curves has drawn wide attention from cryptographic researchers. It is well known that the Weierstrass model provides a general parametrization of elliptic curves. In other words, an elliptic curve over a field K (excluding char$(K) = 2, 3$) is the set of points (x, y) satisfying the equation

$$y^2 = x^3 + ax + b$$

for some $a, b \in K$ where $4a^3 + 27b^2 \neq 0$ together with the point at infinity $\mathcal{O}$. These points exhibit a group structure under an explicitly defined additive group law. In other words, two points $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ can be added to form a third point $R = P + Q = (x_3, y_3)$ on the same curve. The negative of the point P is $(x_1, -y_1)$. The identity element is the point at infinity $\mathcal{O}$. From this we can define a scalar multiple S of a point P as

$$S = [k]P = \underbrace{P + P + \ldots + P}_{k \text{ times}}.$$

Computing k when only P and S are known is believed to be intractable for carefully selected parameters. This forms the basis of the elliptic curve discrete logarithm problem, which is used to provide cryptographic security.

One of the main challenges in elliptic curve cryptography is to perform scalar multiplication efficiently under different environmental constraints (such as resistance to side channel attacks, bandwidth efficiency, memory limitations). In this paper, we restrict attention to the optimization of point addition and point doubling which are vital for the overall performance of double-and-add type scalar multiplication algorithms.

Elliptic curves can be represented in several different ways. To obtain faster group operations, some other elliptic curve representations have also been considered in the last two decades. In this context, we present a short outline of previous work on which our paper is built.

- Chudnovsky & Chudnovsky (1986) developed the first inversion-free algorithms and reported the operation counts for performing arithmetic on Weierstrass, Jacobi quartic, Jacobi intersection, and Hessian curves.
- Cohen et al. (1998) provided efficient strategies for scalar multiplication on Weierstrass curves. Doche et al. (2006) introduced fast doubling and tripling algorithms on Weierstrass curves for two special families. The doubling algorithm in (Doche et al. 2006) was improved by Bernstein et al. (2007) for $\mathbf{S} < \mathbf{M}$.
- In chronological order, Joye & Quisquater (2001), Liardet & Smart (2001), Brier & Joye (2002), Billet & Joye (2003) showed ways of performing scalar multiplication with resistance to side channel attacks using Hessian, Jacobi intersection, Weierstrass and Jacobi quartic forms, respectively.
- Duquesne (2007) proposed a faster algorithm for computing point addition on Jacobi quartic curves based on the formulae in (Billet & Joye 2003) by using an alternative coordinate system. In (Bernstein & Lange 2007*b*) and (Bernstein & Lange 2007*a*) a better operation count for $\mathbf{S} < \mathbf{M}$ was proposed. Some of the optimizations in this paper benefit from similar ideas.
- Bernstein & Lange (2007*c*) introduced Edwards curves for providing fast arithmetic and efficient countermeasures to side channel attacks. Later, Bernstein & Lange (2007*d*) proposed the inverted Edwards coordinates which improve timings for Edwards curves and provided the fastest unified addition of that time. Bernstein & Lange (2007*b*)

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

have built a database of explicit formulae that are reported in the literature together with their own optimizations.

For security considerations, the selected curves should have a small cofactor, typically equal to or less than 4. It is possible to find cryptographically interesting curves which satisfy the security criterion and which can be parameterized by one of the curve models mentioned above. See (Liardet & Smart 2001), (Billet & Joye 2003), and (Bernstein & Lange 2007*c*) for sample curves.

In this work, we aim to speedup the group operations for these curves with a final aim of improving the best timings for various scalar multiplication algorithms. We extend the literature by introducing new addition and doubling formulae for various curve models. An extensive speed comparison is given in the appendix. From the comparison tables it can be observed that most of our optimizations achieve the removal of field multiplications and/or field squarings in comparison to the current literature. In addition, we provide **S**-**M** tradeoffs for the doubling operations.

For a quick reference, we present a snapshot of some of the latest operation counts. We explain these results in detail with necessary pointers to the literature in Section 2. In what follows we will frequently use the terms *unified addition*, *readdition*, and *mixed addition*. Unified addition means that addition formulae remain valid when two input points are same, see (Cohen et al. 2005, Section 29.1.2). Readdition means that a point addition has already taken place and some of the previously computed data is cached, see (Cohen et al. 1998) or (Bernstein & Lange 2007*c*, p.40). Mixed addition means that one of the addends is given in affine coordinates, see (Cohen et al. 1998).

Jacobi quartic	Addition	Doubling
Literature record	8**M**+3**S**+1**D**	3**M**+4**S**
This work	7**M**+3**S**+1**D**	2**M**+5**S**+1**D**
Jacobi intersection	Addition	Doubling
Literature record	13**M**+2**S**+1**D**	3**M**+4**S**
This work	11**M**+1**S**+2**D**	2**M**+5**S**+1**D**
Edwards	Addition	Doubling
Literature record	9**M**+1**S**+1**D**	3**M**+4**S**
This work	11**M**	-
Hessian	Addition	Doubling
Literature record	12**M**	3**M**+6**S**
This work	6**M**+6**S**	3**M**+6**S**

The paper is organized as follows. We provide new formulae and better operation counts for various elliptic curve forms in Section 2. A naming of different systems are pointed in Section 3 The exceptional cases are considered in Section 4. We make comparisons of various systems and draw our conclusions in Section 5.

2 Improvements

In the rest of this paper, we assume K is finite, is of large size, and $\mathrm{char}(K) \neq 2, 3$. For any elliptic curve over K we restrict our attention to the K-rational points. Not all of these assumptions are always necessary. However, they make our investigation easier. We omit the operation counts for affine coordinates since known formulae for this representation require field inversions which are very costly in most implementations compared to field multiplications. We also omit the cost of additions, subtractions, and multiplication by very small constants (e.g. 2, 4, etc.). However, they can be properly counted from the provided algorithms if they are not negligible.

Some of the derivations in this section are aided by the use of (Monagan & Pearce 2006) simplification algorithms for rational expressions. We also use computer aid with Maple v.11[2] computer algebra system. We obtain curve definitions and affine versions of various formulae from (Bernstein & Lange 2007*b*). We borrow the notation **M**, **S**, and **D** from (Bernstein & Lange 2007*c*).

2.1 Jacobi quartic form

The uses of these curves in cryptology are explained by Chudnovsky & Chudnovsky (1986) and Billet & Joye (2003). A Jacobi quartic form elliptic curve over K is defined by $y^2 = x^4 + 2ax^2 + 1$ where $a \in K$ with $a^2 \neq 1$. Birational maps between Weierstrass and Jacobi quartic curves can be found in (Billet & Joye 2003), (Bernstein & Lange 2007*b*), and (Bernstein & Lange 2007*a*).

Our main focus in this work is the group law. Therefore we are interested in explicit formulae which add two points. We use the common notation

$$(x_3, y_3) = (x_1, y_1) + (x_2, y_2)$$

which is used in many textbooks. Here (x_1, y_1) and (x_2, y_2) are the addends and (x_3, y_3) is the sum.

The explicit formulae for the group law on Jacobi quartic form elliptic curves date back to (Jacobi 1829). Even earlier, a formula for computing x_3 (see below in (1)) appears in one of Euler's works from the 18^{th} century, (Euler 1761). A formula for computing y_3 can be found in (McKean & Moll 1927, p.111). We will proceed with working on the affine version of the unified addition formulae in (Billet & Joye 2003) given by

$$(x_3, y_3) = (x_1, y_1) + (x_2, y_2)$$

where

$$\begin{aligned} x_3 &= \frac{x_1y_2 + y_1x_2}{1 - x_1^2x_2^2}, \\ y_3 &= \frac{(y_1y_2 + 2ax_1x_2)(x_1^2x_2^2 + 1) + 2x_1x_2(x_1^2 + x_2^2)}{(1 - x_1^2x_2^2)^2}. \end{aligned} \tag{1}$$

The identity element is the point $(0, 1)$. The negative of a point (x, y) is $(-x, y)$. In this section, we will update the numerator of y_3. If the numerator is designated t then we have

$$\begin{aligned} t &= (y_1y_2 + 2ax_1x_2)(x_1^2x_2^2 + 1) + 2x_1x_2(x_1^2 + x_2^2) \\ &= (y_1y_2 + 2ax_1x_2)(x_1^2x_2^2 + 1) + 2x_1x_2(x_1^2 + x_2^2) + \\ &\quad x_1^2y_2^2 + 2x_1y_1x_2y_2 + y_1^2x_2^2 - (x_1y_2 + y_1x_2)^2. \end{aligned}$$

Using the curve equation $y^2 = x^4 + 2ax^2 + 1$, we replace y_1^2 with $x_1^4 + 2ax_1^2 + 1$ and y_2^2 with $x_2^4 + 2ax_2^2 + 1$. This yields

$$\begin{aligned} t &= (y_1y_2 + 2ax_1x_2)(x_1^2x_2^2 + 1) + 2x_1x_2(x_1^2 + x_2^2) + \\ &\quad x_1^2(x_2^4 + 2ax_2^2 + 1) + 2x_1y_1x_2y_2 + \\ &\quad x_2^2(x_1^4 + 2ax_1^2 + 1) - (x_1y_2 + y_1x_2)^2. \end{aligned}$$

We obtain the new formula for y_3 by organizing the terms. The new unified addition formulae are given by

$$(x_3, y_3) = (x_1, y_1) + (x_2, y_2)$$

where

$$\begin{aligned} x_3 &= \frac{x_1y_2 + y_1x_2}{1 - x_1^2x_2^2}, \\ y_3 &= \left(\frac{x_1x_2 + 1}{1 - x_1^2x_2^2}\right)^2 \Big((x_1^2 + 1)(x_2^2 + 1) + \\ &\quad y_1y_2 + (2a - 2)\,x_1x_2\Big) - x_3^2 - 1. \end{aligned} \tag{2}$$

[2] http://www.maplesoft.com

A projective weighted coordinate systems is used in (Chudnovsky & Chudnovsky 1986) and in (Billet & Joye 2003) for the elimination of field inversions. In this system, each point is represented by the triplet $(X : Y : Z)$ which satisfies the equation $Y^2 = X^4 + 2aX^2Z^2 + Z^4$ and corresponds to the affine point $(X/Z, Y/Z^2)$ with $Z \neq 0$. The identity element is represented by $(0 : 1 : 1)$. The negative of $(X : Y : Z)$ is $(-X : Y : Z)$. The new point addition (2) in projective weighted coordinates then becomes

$$(X_3 : Y_3 : Z_3) = (X_1 : Y_1 : Z_1) + (X_2 : Y_2 : Z_2)$$

where

$$\begin{aligned} X_3 &= X_1Z_1Y_2 + Y_1X_2Z_2, \\ Z_3 &= Z_1^2Z_2^2 - X_1^2X_2^2, \\ Y_3 &= (X_1X_2 + Z_1Z_2)^2((X_1^2 + Z_1^2)(X_2^2 + Z_2^2) + \\ &\quad Y_1Y_2 + (2a - 2)X_1Z_1X_2Z_2) - X_3^2 - Z_3^2. \end{aligned} \tag{3}$$

Rather than using the projective weighted coordinates, we use a redundant representation of points for efficiency purposes. This representation is based on the work of Duquesne (2007) which is extended in (Bernstein & Lange 2007*a*).

We represent a point with $Z \neq 0$ with the sextuplet $(X : Y : Z : X^2 : Z^2 : XZ)$ and incorporate this representation with the new point addition formulae (3). Now, $(X_1 : Y_1 : Z_1 : U_1 : V_1 : W_1)$ and $(X_2 : Y_2 : Z_2 : U_2 : V_2 : W_2)$ with $U_1 = X_1^2$, $V_1 = Z_1^2$, $W_1 = X_1Z_1$, $U_2 = X_2^2$, $V_2 = Z_2^2$, $W_2 = X_2Z_2$ can be added with the algorithm

$$\begin{gathered} A \leftarrow U_1U_2, \quad B \leftarrow V_1V_2, \quad C \leftarrow W_1W_2, \quad D \leftarrow Y_1Y_2, \\ X_3 \leftarrow (W_1 + Y_1)(W_2 + Y_2) - C - D, \quad Z_3 \leftarrow B - A, \\ U_3 \leftarrow X_3^2, \quad V_3 \leftarrow Z_3^2, \quad F \leftarrow A + B + 2C, \\ G \leftarrow (U_1 + V_1)(U_2 + V_2) + kC + D, \quad H \leftarrow U_3 + V_3, \\ Y_3 \leftarrow FG - H, \quad W_3 \leftarrow ((X_3 + Z_3)^2 - H)/2 \end{gathered}$$

where $k = 2(a - 1)$. The new unified addition costs $7\mathbf{M} + 3\mathbf{S} + 1\mathbf{D}$ in the modified coordinates. Assuming that $(X_2 : Y_2 : Z_2 : U_2 : V_2 : W_2)$ is cached, a readdition costs $7\mathbf{M} + 3\mathbf{S} + 1\mathbf{D}$. A $6\mathbf{M} + 3\mathbf{S} + 1\mathbf{D}$ mixed addition can be derived by setting $Z_2 = 1$. We use the name "modified Jacobi quartic v.2b" to refer to this coordinate system. Modified Jacobi quartic v.2b uses the new addition formulae and a $3\mathbf{M} + 4\mathbf{S}$ doubling algorithm proposed by Hisil et al. (2007).

To evaluate the new addition formulae, a similar algorithm for a less redundant version of modified Jacobi quartic v.2b which represents points with the quintuplet $(X : Y : Z : U : V)$, is also very efficient in practice. This point representation is proposed in (Hisil et al. 2007). In this system the new unified addition costs $7\mathbf{M}+4\mathbf{S}+1\mathbf{D}$ (by computing $W_1 = ((X_1 + Z_1)^2 - U_1 - V_1)/2$ and $W_2 = ((X_2 + Z_2)^2 - U_2 - V_2)/2$ on the fly, and not computing W_3). Following this and assuming that $(X_2 : Y_2 : Z_2 : U_2 : V_2)$ is cached, the readdition costs $7\mathbf{M}+3\mathbf{S}+1\mathbf{D}$ with the extra caching of W_2. A $6\mathbf{M} + 3\mathbf{S} + 1\mathbf{D}$ mixed addition can then be derived by setting $Z_2 = 1$. We use the name "modified Jacobi quartic v.2a" to refer to this system. This system also uses $3\mathbf{M} + 4\mathbf{S}$ doubling algorithm in (Hisil et al. 2007). A comparison of our results with the literature is given as follows.

Jacobi quartic	Addition
(Billet & Joye 2003), ($\epsilon = 1$)	$10\mathbf{M}$+$3\mathbf{S}$+$1\mathbf{D}$
(Duquesne 2007), ($\epsilon = 1$)	$9\mathbf{M}$+$2\mathbf{S}$+$1\mathbf{D}$
(Bernstein & Lange 2007*b*)	$8\mathbf{M}$+$3\mathbf{S}$+$1\mathbf{D}$
This work (modified v.2a)	$7\mathbf{M}$+$4\mathbf{S}$+$1\mathbf{D}$
This work (modified v.2b)	$7\mathbf{M}$+$3\mathbf{S}$+$1\mathbf{D}$

It is convenient here to note that the $3\mathbf{M} + 4\mathbf{S}$ doubling algorithm in (Hisil et al. 2007) can be easily derived from the new affine addition formulae (2) as follows. We symbolically input the same points to the new addition formulae and obtain

$$(x_3, y_3) = [2](x_1, y_1)$$

where

$$\begin{aligned} x_3 &= \frac{2x_1y_1}{1 - x_1^4}, \\ y_3 &= \left(\frac{x_1^2 + 1}{1 - x_1^4}\right)^2 (x_1^4 + 2ax_1^2 + 1 + y_1^2) - x_3^2 - 1. \end{aligned} \tag{4}$$

We then replace $x_1^4 + 2ax_1^2 + 1$ with y_1^2 using the curve equation. This yields

$$(x_3, y_3) = [2](x_1, y_1)$$

where

$$\begin{aligned} x_3 &= \frac{2x_1y_1}{1 - x_1^4} \\ y_3 &= 2\left(\frac{y_1(x_1^2 + 1)}{1 - x_1^4}\right)^2 - x_3^2 - 1 \end{aligned} \tag{5}$$

The point doubling formulae (5) in projective weighted coordinates are given by

$$(X_3 : Y_3 : Z_3) = [2](X_1 : Y_1 : Z_1)$$

where

$$\begin{aligned} X_3 &= 2X_1Y_1Z_1, \\ Z_3 &= Z_1^4 - X_1^4, \\ Y_3 &= 2(Y_1(X_1^2 + Z_1^2))^2 - X_3^2 - Z_3^2. \end{aligned} \tag{6}$$

These formulae are advantageous when used with both versions of the modified coordinates. The point doubling algorithm for (6) is given by

$$\begin{gathered} A \leftarrow U_1 + V_1, \quad X_3 \leftarrow 2Y_1W_1, \quad Z_3 \leftarrow A(V_1 - U_1), \\ U_3 \leftarrow X_3^2, \quad V_3 \leftarrow Z_3^2, \quad B \leftarrow U_3 + V_3, \\ W_3 \leftarrow ((X_3 + Z_3)^2 - B)/2, \quad Y_3 \leftarrow 2(Y_1A)^2 - B. \end{gathered}$$

Doubling costs $3\mathbf{M} + 4\mathbf{S}$ in both versions of the modified coordinates. See the works (Hisil et al. 2007) and (Bernstein & Lange 2007*b*).

Building on similar ideas, it is possible to derive the following doubling formulae

$$(x_3, y_3) = [2](x_1, y_1)$$

where

$$\begin{aligned} x_3 &= \frac{2x_1y_1}{1 - x_1^4}, \\ y_3 &= 2\left(\frac{y_1^2}{1 - x_1^4}\right)^2 - ax_3^2 - 1. \end{aligned} \tag{7}$$

The new doubling formulae in projective weighted coordinates are given by

$$(X_3 : Y_3 : Z_3) = [2](X_1 : Y_1 : Z_1)$$

where

$$\begin{aligned} X_3 &= 2X_1Y_1Z_1, \\ Z_3 &= Z_1^4 - X_1^4, \\ Y_3 &= 2Y_1^4 - aX_3^2 - Z_3^2. \end{aligned} \tag{8}$$

These formulae are again advantageous when used with both versions of the modified coordinates. We reproduce both versions of the modified coordinates with the names "modified Jacobi quartic v.3a" and "modified Jacobi quartic v3.b" to emphasize the use of the new doubling formulae together with the new addition formulae (3). A point $(X_1\colon Y_1\colon Z_1\colon U_1\colon V_1\colon W_1)$ can be doubled with the algorithm

$$X_3 \leftarrow 2Y_1W_1, \quad Z_3 \leftarrow (V_1 - U_1)(V_1 + U_1), \quad U_3 \leftarrow X_3^2,$$
$$V_3 \leftarrow Z_3^2, \quad W_3 \leftarrow ((X_3 + Z_3)^2 - U_3 - V_3)/2,$$
$$Y_3 \leftarrow 2Y_1^4 - aU_3 - V_3.$$

Doubling costs $2\mathbf{M} + 5\mathbf{S} + 1\mathbf{D}$ in both versions of the modified coordinates. A comparison of our results with the literature is given as follows. The operation counts from the first three entries are from (Bernstein & Lange 2007*b*).

Jacobi quartic	Doubling
Bernstein/Lange "dbl-2007-bl"	$1\mathbf{M}+9\mathbf{S}+1\mathbf{D}$
Hisil/Carter/Dawson "dbl-2007-hcd"	$2\mathbf{M}+6\mathbf{S}+2\mathbf{D}$
Feng/Wu "dbl-2007-fw-4"	$8\mathbf{S}+3\mathbf{D}$
(Hisil et al. 2007)	$3\mathbf{M}+4\mathbf{S}$
This work	$2\mathbf{M}+5\mathbf{S}+1\mathbf{D}$

For further comparison, see modified Jacobi quartic v.2a, modified Jacobi quartic v2.b, modified Jacobi quartic v.3a, and modified Jacobi quartic v3.b in the appendix.

2.2 Jacobi intersection form

The uses of these curves in cryptology are explained by Chudnovsky & Chudnovsky (1986) and Liardet & Smart (2001). The explicit formulae for the group law date back to (Jacobi 1829). A Jacobi intersection form elliptic curve over K is defined by

$$\begin{cases} s^2 + c^2 = 1 \\ as^2 + d^2 = 1 \end{cases}$$

where $a \in K$ with $a(1-a) \neq 0$. Birational maps between Weierstrass and Jacobi intersection curves can be found in (Liardet & Smart 2001), (Bernstein & Lange 2007*b*), and (Bernstein & Lange 2007*a*). Following the notation of (Chudnovsky & Chudnovsky 1986), the affine version of the unified addition formulae are given by

$$(s_3, c_3, d_3) = (s_1, c_1, d_1) + (s_2, c_2, d_2)$$

where

$$\begin{aligned} s_3 &= \frac{s_1c_2d_2 + c_1d_1s_2}{c_2^2 + d_1^2s_2^2}, \\ c_3 &= \frac{c_1c_2 - s_1d_1s_2d_2}{c_2^2 + d_1^2s_2^2}, \\ d_3 &= \frac{d_1d_2 - as_1c_1s_2c_2}{c_2^2 + d_1^2s_2^2}. \end{aligned} \tag{9}$$

The identity element is the point $(0, 1, 1)$. The negative of a point (s, c, d) is $(-s, c, d)$. Chudnovsky & Chudnovsky (1986) use projective homogenous coordinates to eliminate field inversions. In this system, each point is represented by the quadruplet $(S\colon C\colon D\colon T)$ which satisfies the equations $S^2 + C^2 = T^2$ and $aS^2 + D^2 = T^2$ simultaneously and corresponds to the affine point $(S/T, C/T, D/T)$ with $T \neq 0$. The identity element is represented by $(0\colon 1\colon 1\colon 1)$. The negative of $(S\colon C\colon D\colon T)$ is $(-S\colon C\colon D\colon T)$. The point addition (9) in projective homogenous coordinates is given by

$$(S_3\colon C_3\colon D_3\colon T_3) = (S_1\colon C_1\colon D_1\colon T_1) + (S_2\colon C_2\colon D_2\colon T_2)$$

where

$$\begin{aligned} S_3 &= S_1T_1C_2D_2 + C_1D_1S_2T_2, \\ C_3 &= C_1T_1C_2T_2 - S_1D_1S_2D_2, \\ D_3 &= D_1T_1D_2T_2 - aS_1C_1S_2C_2, \\ T_3 &= D_1^2S_2^2 + T_1^2C_2^2. \end{aligned} \tag{10}$$

To eliminate several field multiplications, we modify the homogenous projective coordinates where each point is represented by the sextuplet, $(S\colon C\colon D\colon T\colon SC\colon DT)$. The points represented by $(S_1\colon C_1\colon D_1\colon T_1\colon U_1\colon V_1)$ and $(S_2\colon C_2\colon D_2\colon T_2\colon U_2\colon V_2)$ with $U_1 = S_1C_1$, $V_1 = D_1T_1$, $U_2 = S_2C_2$, $V_2 = D_2T_2$ can be added with the algorithm

$$E \leftarrow S_1D_2, \quad F \leftarrow C_1T_2, \quad G \leftarrow D_1S_2, \quad H \leftarrow T_1C_2,$$
$$J \leftarrow U_1V_2, \quad K \leftarrow V_1U_2, \quad S_3 \leftarrow (H + F)(E + G) - J - K,$$
$$C_3 \leftarrow (H + E)(F - G) - J + K,$$
$$D_3 \leftarrow (V_1 - aU_1)(U_2 + V_2) + aJ - K,$$
$$T_3 \leftarrow (H + G)^2 - 2K, \quad U_3 \leftarrow S_3C_3, \quad V_3 \leftarrow D_3T_3.$$

The unified point addition costs $11\mathbf{M} + 1\mathbf{S} + 2\mathbf{D}$ in the modified coordinates. Assuming that $(S_2\colon C_2\colon D_2\colon T_2\colon U_2\colon V_2)$ is cached, the readdition costs $11\mathbf{M} + 1\mathbf{S} + 2\mathbf{D}$. A $10\mathbf{M} + 1\mathbf{S} + 2\mathbf{D}$ mixed addition is easily derived by setting $T_2 = 1$. We use the name "modified Jacobi intersection" to refer to this system.

A similar algorithm can be used for projective homogenous coordinates. The unified addition costs $13\mathbf{M} + 1\mathbf{S} + 2\mathbf{D}$ computing $U_1 = S_1C_1$, $V_1 = D_1T_1$, $U_2 = S_2C_2$, $V_2 = D_2T_2$ on the fly, and not computing U_3 and V_3. Following this and assuming that $(S_2\colon C_2\colon D_2\colon T_2)$ is cached, the readdition costs $11\mathbf{M} + 1\mathbf{S} + 2\mathbf{D}$ with the extra caching of U_2 and V_2. A $10\mathbf{M} + 1\mathbf{S} + 2\mathbf{D}$ mixed addition is then derived by setting $T_2 = 1$. We use the name "Jacobi intersection v.2" to refer to this system which uses the new addition algorithm. A comparison of our results with the literature is given as follows.

Jacobi intersection	Addition
(Chudnovsky & Chudnovsky 1986)	$14\mathbf{M}+2\mathbf{S}+1\mathbf{D}$
(Liardet & Smart 2001)	$13\mathbf{M}+2\mathbf{S}+1\mathbf{D}$
This work (projective)	$13\mathbf{M}+1\mathbf{S}+2\mathbf{D}$
This work (modified)	$11\mathbf{M}+1\mathbf{S}+2\mathbf{D}$

Efficient doubling formulae for the modified Jacobi intersection coordinates can be derived starting from the unified addition formulae (10). We symbolically input the same points into the original addition formulae and obtain

$$(s_3, c_3, d_3) = [2](s_1, c_1, d_1)$$

where

$$\begin{aligned} s_3 &= \frac{2s_1c_1d_1}{c_1^2 + s_1^2d_1^2}, \\ c_3 &= \frac{c_1^2 - s_1^2d_1^2}{c_1^2 + s_1^2d_1^2}, \\ d_3 &= \frac{d_1^2 - as_1^2c_1^2}{c_1^2 + s_1^2d_1^2}. \end{aligned} \tag{11}$$

Using the defining equations, $s^2 + c^2 = 1$ and $as^2 + d^2 = 1$, we replace c_1^2 with $c_1^2(as_1^2 + d_1^2)$ (only for the denominators) and $s_1^2d_1^2$ with $(1 - c_1^2)d_1^2$. This yields

$$\begin{aligned} s_3 &= (2s_1c_1d_1)/(c_1^2(as_1^2 + d_1^2) + (1 - c_1^2)d_1^2), \\ c_3 &= (c_1^2(as_1^2 + d_1^2) - (1 - c_1^2)d_1^2)/(c_1^2(as_1^2 + d_1^2) + (1 - c_1^2)d_1^2), \\ d_3 &= (d_1^2 - as_1^2c_1^2)/(c_1^2(as_1^2 + d_1^2) + (1 - c_1^2)d_1^2). \end{aligned}$$

These substitutions give an intermediate formula for c_3 where

$$\begin{aligned} s_3 &= (2s_1c_1d_1)/(d_1^2+as_1^2c_1^2), \\ c_3 &= (as_1^2c_1^2+2c_1^2d_1^2-d_1^2)/(d_1^2+as_1^2c_1^2), \\ d_3 &= (d_1^2-as_1^2c_1^2)/(d_1^2+as_1^2c_1^2). \end{aligned}$$

Finally, we replace $2c_1^2d_1^2$ with $2c_1^2(s_1^2+c_1^2-as_1^2)$ in c_3.

$$\begin{aligned} s_3 &= (2s_1c_1d_1)/(as_1^2c_1^2+d_1^2), \\ c_3 &= (as_1^2c_1^2+2c_1^2(s_1^2+c_1^2-as_1^2)-d_1^2)/(as_1^2c_1^2+d_1^2), \\ d_3 &= (d_1^2-as_1^2c_1^2)/(as_1^2c_1^2+d_1^2). \end{aligned}$$

The new following doubling formulae are given by

$$(s_3,c_3,d_3)=[2](s_1,c_1,d_1)$$

where

$$\begin{aligned} s_3 &= \frac{2s_1c_1d_1}{d_1^2+as_1^2c_1^2}, \\ c_3 &= \frac{-d_1^2-as_1^2c_1^2+2(s_1^2c_1^2+c_1^4)}{d_1^2+as_1^2c_1^2}, \\ d_3 &= \frac{d_1^2-as_1^2c_1^2}{d_1^2+as_1^2c_1^2}. \end{aligned} \tag{12}$$

The new doubling formulae (12) in projective homogenous coordinates are given by

$$(S_3\colon C_3\colon D_3\colon T_3)=[2](S_1\colon C_1\colon D_1\colon T_1)$$

where

$$\begin{aligned} S_3 &= 2S_1C_1D_1T_1, \\ C_3 &= -D_1^2T_1^2-aS_1^2C_1^2+2(S_1^2C_1^2+C_1^4), \\ D_3 &= D_1^2T_1^2-aS_1^2C_1^2, \\ T_3 &= D_1^2T_1^2+aS_1^2C_1^2. \end{aligned} \tag{13}$$

Now, $(S_1\colon C_1\colon D_1\colon T_1\colon U_1\colon V_1)$ can be doubled with the algorithm

$$E \leftarrow V_1^2, \quad F \leftarrow U_1^2, \quad G \leftarrow aF, \quad T_3 \leftarrow E+G,$$
$$D_3 \leftarrow E-G, \quad C_3 \leftarrow 2(F+C_1^4)-T_3,$$
$$S_3 \leftarrow (U_1+V_1)^2-E-F, \quad U_3 \leftarrow S_3C_3, \quad V_3 \leftarrow D_3T_3.$$

It is easy to see that point doubling costs $2\mathbf{M}+5\mathbf{S}+1\mathbf{D}$ both on projective homogenous and modified projective homogenous coordinates. A comparison of our results with the literature is given as follows.

Jacobi intersection	Doubling
(Liardet & Smart 2001)	$4\mathbf{M}+3\mathbf{S}$
(Bernstein & Lange 2007*b*)	$3\mathbf{M}+4\mathbf{S}$
This work	$2\mathbf{M}+5\mathbf{S}+1\mathbf{D}$

2.3 Edwards form

The uses of these curves in cryptology are explained by Bernstein & Lange (2007*c*), Bernstein et al. (2007), and Bernstein & Lange (2007*d*). An Edwards form elliptic curve over K is defined by $x^2+y^2=c^2(1+dx^2y^2)$ where $c,d\in K$ with $cd(1-c^4d)\neq 0$. Birational maps between Weierstrass and Edwards curves are given by (Bernstein & Lange 2007*c*). The affine unified addition formulae are given by

$$(x_3,y_3)=(x_1,y_1)+(x_2,y_2)$$

where

$$\begin{aligned} x_3 &= \frac{x_1y_2+y_1x_2}{c(1+dx_1y_1x_2y_2)}, \\ y_3 &= \frac{y_1y_2-x_1x_2}{c(1-dx_1y_1x_2y_2)}. \end{aligned} \tag{14}$$

The identity element is the point $(0,c)$. The negative of a point (x,y) is $(-x,y)$. We first describe how new addition formulae for Edwards curves can be derived from the original addition formulae in (Bernstein & Lange 2007*c*). Consider the relations $x_1^2+y_1^2-c^2(1+dx_1^2y_1^2)=0$, $x_2^2+y_2^2-c^2(1+dx_2^2y_2^2)=0$ obtained from the curve equation. From this, we can express c and d in terms of x_1,x_2,y_1,y_2 as follows.

$$\begin{aligned} c^2 &= \frac{x_1^2x_2^2y_1^2-x_1^2x_2^2y_2^2+x_1^2y_1^2y_2^2-x_2^2y_1^2y_2^2}{x_1^2y_1^2-x_2^2y_2^2}, \\ d &= \frac{x_1^2-x_2^2+y_1^2-y_2^2}{x_1^2x_2^2y_1^2-x_1^2x_2^2y_2^2+x_1^2y_1^2y_2^2-x_2^2y_1^2y_2^2}. \end{aligned}$$

Substitutions can be made in the original addition formulae to obtain

$$\begin{aligned} x_3 = {} & (x_1y_2+y_1x_2)/((1/c)(x_1^2x_2^2y_1^2-x_1^2x_2^2y_2^2+x_1^2y_1^2y_2^2- \\ & x_2^2y_1^2y_2^2)/(x_1^2y_1^2-x_2^2y_2^2)(1+(x_1^2-x_2^2+y_1^2-y_2^2)/ \\ & (x_1^2x_2^2y_1^2-x_1^2x_2^2y_2^2+x_1^2y_1^2y_2^2-x_2^2y_1^2y_2^2)x_1y_1x_2y_2)), \\ y_3 = {} & (y_1y_2-x_1x_2)/((1/c)(x_1^2x_2^2y_1^2-x_1^2x_2^2y_2^2+x_1^2y_1^2y_2^2- \\ & x_2^2y_1^2y_2^2)/(x_1^2y_1^2-x_2^2y_2^2)(1-(x_1^2-x_2^2+y_1^2-y_2^2)/ \\ & (x_1^2x_2^2y_1^2-x_1^2x_2^2y_2^2+x_1^2y_1^2y_2^2-x_2^2y_1^2y_2^2)x_1y_1x_2y_2)). \end{aligned}$$

After straightforward simplifications, the new addition formulae are given by

$$(x_3,y_3)=(x_1,y_1)+(x_2,y_2)$$

where

$$\begin{aligned} x_3 &= \frac{c(x_1y_1+x_2y_2)}{x_1x_2+y_1y_2}, \\ y_3 &= \frac{c(x_1y_1-x_2y_2)}{x_1y_2-y_1x_2}. \end{aligned} \tag{15}$$

Note, the formula for computing y_3 is not defined for $(x_1,y_1)=(x_2,y_2)$ and hence this addition is not unified. For this reason, we call the new formulae *dedicated* addition for Edwards curves. These new formulae show an interesting fact that dedicated addition on the Edwards curves does not depend on the curve parameter d. Therefore, arbitrary selections of d do not cause any efficiency loss.

Bernstein & Lange (2007*c*) use homogenous projective coordinates to prevent field inversions that appear in the affine formulae. We also represent each point in projective homogenous coordinates for the new formulae (15). Each point is represented by the triplet $(X\colon Y\colon Z)$ which satisfies the projective curve $(X^2+Y^2)Z^2=c^2(Z^4+dX^2Y^2)$ and corresponds to the affine point $(X/Z,Y/Z)$ with $Z\neq 0$. The identity element is represented by $(0\colon c\colon 1)$. The negative of $(X\colon Y\colon Z)$ is $(-X\colon Y\colon Z)$. The new addition formulae in projective homogenous coordinates are given by

$$(X_3\colon Y_3\colon Z_3)=(X_1\colon Y_1\colon Z_1)+(X_2\colon Y_2\colon Z_2)$$

where

$$\begin{aligned} X_3 &= Z_1Z_2(X_1Y_2-Y_1X_2)(X_1Y_1Z_2^2+Z_1^2X_2Y_2), \\ Y_3 &= Z_1Z_2(X_1X_2+Y_1Y_2)(X_1Y_1Z_2^2-Z_1^2X_2Y_2), \\ Z_3 &= kZ_1^2Z_2^2(X_1X_2+Y_1Y_2)(X_1Y_2-Y_1X_2) \end{aligned} \tag{16}$$

with $k=1/c$. Now, $(X_1\colon Y_1\colon Z_1)$ and $(X_2\colon Y_2\colon Z_2)$ can be added with the algorithm

$$A \leftarrow X_1Z_2, \quad B \leftarrow Y_1Z_2, \quad C \leftarrow Z_1X_2, \quad D \leftarrow Z_1Y_2,$$
$$E \leftarrow AB, \quad F \leftarrow CD, \quad G \leftarrow E+F, \quad H \leftarrow E-F,$$
$$J \leftarrow (A-C)(B+D)-H, \quad K \leftarrow (A+D)(B+C)-G,$$
$$X_3 \leftarrow GJ, \quad Y_3 \leftarrow HK, \quad Z_3 \leftarrow k\,JK.$$

We also investigate the case for inverted Edwards coordinates introduced by Bernstein & Lange (2007*d*). In this system, each triplet $(X\colon Y\colon Z)$ satisfies the curve $(X^2+Y^2)Z^2 = c^2(dZ^4+X^2Y^2)$ and corresponds to the affine point $(Z/X, Z/Y)$ with $XYZ \neq 0$. The identity element is represented by the vector $(c,0,0)$. The negative of $(X\colon Y\colon Z)$ is $(-X\colon Y\colon Z)$. The new addition formulae (15) in inverted Edwards coordinates are given by

$$(X_3\colon Y_3\colon Z_3) = (X_1\colon Y_1\colon Z_1) + (X_2\colon Y_2\colon Z_2)$$

where

$$\begin{aligned} X_3 &= Z_1Z_2(X_1X_2+Y_1Y_2)(X_1Y_1Z_2^2 - Z_1^2X_2Y_2),\\ Y_3 &= Z_1Z_2(X_1Y_2-Y_1X_2)(X_1Y_1Z_2^2 + Z_1^2X_2Y_2),\\ Z_3 &= c(X_1Y_1Z_2^2 + Z_1^2X_2Y_2)(X_1Y_1Z_2^2 - Z_1^2X_2Y_2). \end{aligned} \tag{17}$$

$(X_1\colon Y_1\colon Z_1)$ and $(X_2\colon Y_2\colon Z_2)$ can be added with the algorithm

$$\begin{gathered} A \leftarrow X_1Z_2, \quad B \leftarrow Y_1Z_2, \quad C \leftarrow Z_1X_2, \quad D \leftarrow Z_1Y_2,\\ E \leftarrow AB, \quad F \leftarrow CD, \quad G \leftarrow E+F, \quad H \leftarrow E-F,\\ X_3 \leftarrow ((A+D)(B+C)-G)H,\\ Y_3 \leftarrow ((A-C)(B+D)-H)G, \quad Z_3 \leftarrow cGH. \end{gathered}$$

A detail to mention is the readdition in projective homogenous coordinates. At this stage, it is more convenient to divide each coordinate of the new formulae (16) by Z_1Z_2. The readdition of $(X_2\colon Y_2\colon Z_2)$ can then be performed with the cached values $R_1 = X_2Y_2$ and $R_2 = Z_2^2$ using the algorithm

$$\begin{gathered} A \leftarrow X_1Y_1, \quad B \leftarrow Z_1^2, \quad C \leftarrow R_2A, \quad D \leftarrow R_1B,\\ E \leftarrow (X_1-X_2)(Y_1+Y_2) - A + R_1,\\ F \leftarrow (X_1+Y_2)(Y_1+X_2) - A - R_1,\\ G \leftarrow ((Z_1+Z_2)^2 - B - R_2)/2, \quad X_3 \leftarrow E(C+D),\\ Y_3 \leftarrow F(C-D), \quad Z_3 \leftarrow kEFG. \end{gathered}$$

In the rest of this section, we assume $c = 1$. See (Bernstein & Lange 2007*c*, Section 4). The dedicated addition then costs $11\mathbf{M}$ for both coordinate systems. A $9\mathbf{M}$ mixed addition can be derived by setting $Z_2 = 1$ again for both coordinate systems. The readdition costs $9\mathbf{M}+2\mathbf{S}$ in projective homogenous coordinates. A comparison of our results with the literature is given as follows.

Edwards (projective)	Addition
(Bernstein & Lange 2007*c*)	10**M**+1**S**+1**D**
This work	11**M**
Edwards (projective)	Readdition
(Bernstein & Lange 2007*c*)	10**M**+1**S**+1**D**
This work	9**M**+2**S**
Edwards (projective)	Mixed addition
(Bernstein & Lange 2007*c*)	9**M**+1**S**+1**D**
This work	9**M**

See "Edwards v.2" in Table 1 and Table 2 in the appendix for further comparison.

In fact, the readdition algorithm shows that a modified version of the homogenous projective Edwards coordinates in which the points are represented by the quintuplet $(X\colon Y\colon Z\colon Z^2\colon XY)$ permits an inversion-free addition in $9\mathbf{M}+2\mathbf{S}$ using the same algorithm. For $\mathbf{S} < \mathbf{M}$, this is faster than the $11\mathbf{M}$ algorithm that we have just described. However, the $3\mathbf{M}+4\mathbf{S}$ doubling algorithm in (Bernstein & Lange 2007*c*) seems to cost $5\mathbf{M}+2\mathbf{S}$ in this coordinate system and also the mixed addition costs $8\mathbf{M}+2\mathbf{S}$ which is slower than the $9\mathbf{M}$ mixed addition given above. Therefore, we do not further consider this system.

The new addition and its associated readdition in inverted Edwards coordinates are not as advantageous as they are for the homogenous projective Edwards coordinates. On the other hand, the mixed addition can be used in some cases. A comparison of the proposed mixed addition with the literature is given as follows.

Edwards (inverted)	Mixed addition
(Bernstein & Lange 2007*d*)	8**M**+1**S**+1**D**
This work	9**M**

See "Inverted Edwards v.2" in Table 1 and Table 2 in the appendix.

We also refer the reader to (Bernstein et al. 2008). We should note here that our more recent work (Hisil et al. 2008) which was published before this work, further improves these operation counts on twisted Edwards curves.

2.4 Hessian form

The uses of these curves in cryptology are explained by Chudnovsky & Chudnovsky (1986), Joye & Quisquater (2001), and Smart (2001). An elliptic curve over K in Hessian form is defined by $x^3+y^3+1 = 3dxy$ where $d \in K$ with $d^3 \neq 1$. Birational maps between Weierstrass and Hessian curves can be found in (Smart 2001), (Joye & Quisquater 2001), (Bernstein & Lange 2007*b*), and (Bernstein & Lange 2007*a*). The addition formulae attributed to Sylvester in (Chudnovsky & Chudnovsky 1986, pp.424-425) are given in (Bernstein & Lange 2007*b*) by

$$(x_3, y_3) = (x_1, y_1) + (x_2, y_2)$$

where

$$\begin{aligned} x_3 &= \frac{y_1^2x_2 - y_2^2x_1}{x_2y_2 - x_1y_1},\\ y_3 &= \frac{x_1^2y_2 - x_2^2y_1}{x_2y_2 - x_1y_1}. \end{aligned} \tag{18}$$

The identity element is the point at infinity. The negative of a point (x,y) is (y,x). On projective homogenous coordinates, each point is represented by the triplet $(X\colon Y\colon Z)$ which satisfies the projective curve $X^3+Y^3+Z^3 = 3dXYZ$ and corresponds to the affine point $(X/Z, Y/Z)$ with $Z \neq 0$. The identity element is represented by $(-1\colon 1\colon 0)$. The negative of $(X\colon Y\colon Z)$ is $(Y\colon X\colon Z)$. The point addition (22) formulae (with each coordinate multiplied by 2) in projective homogenous coordinates are given by,

$$(X_3\colon Y_3\colon Z_3) = (X_1\colon Y_1\colon Z_1) + (X_2\colon Y_2\colon Z_2)$$

where

$$\begin{aligned} X_3 &= 2Y_1^2X_2Z_2 - 2X_1Z_1Y_2^2,\\ Y_3 &= 2X_1^2Y_2Z_2 - 2Y_1Z_1X_2^2,\\ Z_3 &= 2Z_1^2X_2Y_2 - 2X_1Y_1Z_2^2. \end{aligned} \tag{19}$$

The point addition algorithms in (Chudnovsky & Chudnovsky 1986) and (Joye & Quisquater 2001) require $12\mathbf{M}$. To gain speedup in the case $\mathbf{S} < \mathbf{M}$, we modify projective homogenous coordinates with a more redundant representation of points using the nonuplet, $(X\colon Y\colon Z\colon X^2\colon Y^2\colon Z^2\colon 2XY\colon 2XZ\colon 2YZ)$. Two distinct points represented by

$$(X_1\colon Y_1\colon Z_1\colon R_1\colon S_1\colon T_1\colon U_1\colon V_1\colon W_1)$$

and

$$(X_2\colon Y_2\colon Z_2\colon R_2\colon S_2\colon T_2\colon U_2\colon V_2\colon W_2)$$

with $R_1 = X_1^2$, $S_1 = Y_1^2$, $T_1 = Z_1^2$, $U_1 = 2X_1Y_1$, $V_1 = 2X_1Z_1$, $W_1 = 2Y_1Z_1$, $R_2 = X_2^2$, $S_2 = Y_2^2$, $T_2 = Z_2^2$, $U_2 = 2X_2Y_2$, $V_2 = 2X_2Z_2$, $W_2 = 2Y_2Z_2$ can be added with the algorithm

$$X_3 \leftarrow S_1V_2 - V_1S_2, \quad Y_3 \leftarrow R_1W_2 - W_1R_2,$$
$$Z_3 \leftarrow T_1U_2 - U_1T_2, \quad R_3 \leftarrow X_3^2, \quad S_3 \leftarrow Y_3^2, \quad T_3 \leftarrow Z_3^2,$$
$$U_3 \leftarrow (X_3+Y_3)^2 - R_3 - S_3, \quad V_3 \leftarrow (X_3+Z_3)^2 - R_3 - T_3,$$
$$W_3 \leftarrow (Y_3+Z_3)^2 - S_3 - T_3.$$

The addition[3] costs $6\mathbf{M}+6\mathbf{S}$ in the modified Hessian coordinates. Assuming that

$$(X_2\colon Y_2\colon Z_2\colon R_2\colon S_2\colon T_2\colon U_2\colon V_2\colon W_2)$$

is cached, the readdition costs $6\mathbf{M}+6\mathbf{S}$. A $5\mathbf{M}+6\mathbf{S}$ mixed addition can then be derived by setting $Z_2 = 1$. We use the name "modified Hessian" to refer to these results in Section 5. A comparison of our results with the literature is given as follows.

Hessian	Addition
(Chudnovsky & Chudnovsky 1986)	12**M**
(Joye & Quisquater 2001)	12**M**
This work	6**M**+6**S**

A similar algorithm can be used for the homogenous projective coordinates for the readdition and the mixed addition. Assuming that $(X_2\colon Y_2\colon Z_2)$ is cached, the readdition costs $6\mathbf{M}+6\mathbf{S}$ with the extra caching of $R_2, S_2, T_2, U_2, V_2, W_2$. A $5\mathbf{M}+6\mathbf{S}$ mixed addition can be derived by setting $Z_2 = 1$. We use the name "Hessian v.2" to refer to these results in Section 5. Also see (Hisil et al. 2007, pp.146–147).

For speed oriented implementations, Sylvester's doubling formulae are given by

$$(x_3, y_3) = [2](x_1, y_1)$$

where

$$\begin{aligned} x_3 &= \frac{y_1(1-x_1^3)}{x_1^3 - y_1^3}, \\ y_3 &= -\frac{x_1(1-y_1^3)}{x_1^3 - y_1^3}. \end{aligned} \tag{20}$$

When working with the modified coordinates, there exists a doubling strategy which requires no additional effort for generating the new coordinates. The doubling formulae (20) (with each coordinate multiplied by 4) in projective homogenous coordinates are given by

$$\begin{aligned} X_3 &= (2X_1Y_1 - 2Y_1Z_1)(2X_1Z_1 + 2(X_1^2 + Z_1^2)), \\ Y_3 &= (2X_1Z_1 - 2X_1Y_1)(2Y_1Z_1 + 2(Y_1^2 + Z_1^2)), \\ Z_3 &= (2Y_1Z_1 - 2X_1Z_1)(2X_1Y_1 + 2(X_1^2 + Y_1^2)). \end{aligned} \tag{21}$$

Now, $(X_1\colon Y_1\colon Z_1\colon R_1\colon S_1\colon T_1\colon U_1\colon V_1\colon W_1)$ can be doubled with the algorithm

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

[3]Point doubling can be performed after a suitable permutation of coordinates as follows $(Z_1\colon X_1\colon Y_1\colon T_1\colon R_1\colon S_1\colon V_1\colon W_1\colon U_1) + (Y_1\colon Z_1\colon X_1\colon S_1\colon T_1\colon R_1\colon W_1\colon U_1\colon V_1)$ using the addition formulae in the modified Hessian coordinates. This strategy which provides unification of the addition formulae, originates from (Joye & Quisquater 2001, p.6).

Point doubling costs $3\mathbf{M}+6\mathbf{S}$ in both homogenous projective and modified projective homogenous coordinates. A comparison of our results with the literature is given as follows.

Hessian	Doubling
(Chudnovsky & Chudnovsky 1986)	6**M**+3**S**
(Hisil et al. 2007)	7**M**+1**S**
(Hisil et al. 2007)	3**M**+6**S**
This work	3**M**+6**S**

We comment that it is possible to derive unified addition formulae which do not require any permutations of the coordinates to perform doubling. Assuming[4] $x_1x_2 \neq y_1y_2$, we multiply the numerator and the denominator of Sylvester's addition formulae for x_3 by $(x_1^3x_2^3 - y_1^3y_2^3)$ and obtain

$$x_3 = \frac{(x_1^3x_2^3 - y_1^3y_2^3)(y_1^2x_2 - y_2^2x_1)}{(x_1^3x_2^3 - y_1^3y_2^3)(x_2y_2 - x_1y_1)}.$$

This yields

$$\begin{aligned} x_3 = {} & (x_1y_1^2(y_2^3 + x_2^3)(y_2^2y_1 + x_1^2x_2) - \\ & x_2y_2^2(y_1^3 + x_1^3)(y_1^2y_2 + x_2^2x_1))/ \\ & ((x_1^3x_2^3 - y_1^3y_2^3)(x_2y_2 - x_1y_1)). \end{aligned}$$

Using the curve equation $x^2 + y^2 + 1 = 3dxy$, the above expression can be rewritten as

$$\begin{aligned} x_3 = {} & (x_1y_1^2(3dx_2y_2 - 1)(y_2^2y_1 + x_1^2x_2) - \\ & x_2y_2^2(3dx_1y_1 - 1)(y_1^2y_2 + x_2^2x_1))/ \\ & ((x_1^3x_2^3 - y_1^3y_2^3)(x_2y_2 - x_1y_1)). \end{aligned}$$

The numerator can be factorized and cancels with $(x_2y_2 - x_1y_1)$ in the denominator, giving the new addition formulae. The corresponding formula for y_3 can be similarly derived from symmetry. We then have

$$(x_3, y_3) = (x_1, y_1) + (x_2, y_2)$$

where

$$\begin{aligned} x_3 &= \frac{x_1x_2(x_1y_1 + x_2y_2 - 3dx_1x_2y_1y_2) + y_1^2y_2^2}{x_1^3x_2^3 - y_1^3y_2^3}, \\ y_3 &= -\frac{y_1y_2(x_1y_1 + x_2y_2 - 3dx_1x_2y_1y_2) + x_1^2x_2^2}{x_1^3x_2^3 - y_1^3y_2^3}. \end{aligned} \tag{22}$$

The new addition formulae on the projective coordinates are given by

$$\begin{aligned} X_3 = {} & X_1X_2(X_1Y_1Z_2^2 + X_2Y_2Z_1^2 - 3dX_1Y_1X_2Y_2) + \\ & Y_1^2Z_1Y_2^2Z_2, \\ Y_3 = {} & -Y_1Y_2(X_1Y_1Z_2^2 + X_2Y_2Z_1^2 - 3dX_1Y_1X_2Y_2) - \\ & X_1^2Z_1X_2^2Z_2, \\ Z_3 = {} & X_1^3X_2^3 - Y_1^3Y_2^3. \end{aligned}$$

We again use a modified version of the standard coordinates. Two points $(X_1\colon Y_1\colon Z_1\colon V_1\colon W_1)$ and $(X_2\colon Y_2\colon Z_2\colon V_2\colon W_2)$ with $V_1 = X_1Y_1$, $W_1 = Z_1^2$, $V_2 = X_2Y_2$, $W_2 = Z_2^2$ can be added with the algorithm

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

This strategy costs $9\mathbf{M}+6\mathbf{S}+1\mathbf{D}$ which is faster than the unified addition in Weierstrass form in (Brier

[4]This is equivalent to saying $(x_1, y_1) \neq -(x_2, y_2)$. The contrary case should be handled separately as explained in Section 4.

& Joye 2002). However, it is slower than all other unified additions considered in this paper. In addition, doubling, readdition and mixed addition formulae that can be derived from these formulae are not attractive. Therefore, we omit these formulae from further comparison with other systems. We are continuing our search to find other unified addition formulae which can be faster than the proposed formulae.

3 Naming of different systems

The descriptions of systems which are not defined so far (e.g. Jacobi quartic v.1a), can be found in the appendix with the references.

4 Handling exceptional cases

An elliptic curve which can be written in one of these forms always has points of small order (other than the identity). The arithmetic of these points can cause division by zero exceptions when affine formulae are used. These exceptional cases should be handled separately. These cases sometimes require logical checks in the projective representations as well.

Cryptographic applications typically use a large prime order subgroup in which these points (except the identity element, $\mathcal{O}$) do not exist. If this is the case, an implementer only needs to be careful about the identity element. When the points P and Q are to be added, a general strategy to handle the exceptional cases as follows. Let R be the sum of P and Q with $P \neq Q$. Then, $R = Q$ if $P = \mathcal{O}$; $R = P$ if $Q = \mathcal{O}$; $R = \mathcal{O}$ if $P = -Q$. For all other inputs, the sum can be computed with the relevant formulae given in Section 2. Restricting attention to a large prime order subgroup, there are some formulae and coordinate system combinations which do not cause any exception. These are Edwards v.1a, v.1b, Jacobi quartic v.1a, v.1b, Jacobi intersection v.1, v.2, modified Jacobi quartic v.1, v.2a, v.2b, v.3a, v.3b and modified Jacobi intersection. Note, for Edwards v.1a and v.1b, the algorithms work for the whole group of points (i.e. complete) if d is a nonsquare in K. This result is from (Bernstein & Lange 2007*c*). Again restricting attention to a large prime order subgroup, the systems which need logical checks are inverted Edwards (as explained in (Bernstein & Lange 2007*d*)) v.1, v.2, Edwards v.2, Hessian v.1, v.2, and modified Hessian.

5 Comparison and conclusion

There are several scalar multiplication algorithms which can benefit from the optimizations in this paper. We only make comparisons for the popular scalar multiplication strategies using popular elliptic curve parameterizations. We exclude the cost of the final conversion to affine coordinates from our estimations.

Resource limited environments. In memory limited environments (such as smartcards), there is not enough space for storing precomputation tables. For these environments, scalar multiplication with the *"Non-adjacent form without precomputation"* method can be a convenient selection. This algorithm requires 1 doubling, 1/3 mixed addition per scalar bit. The cost estimates are depicted in Table 1.

For example, the best timings for 256-bit scalar multiplication ($\mathbf{S}/\mathbf{M} = 0.8, \mathbf{D}/\mathbf{M} \approx 0$) are obtained by modified Jacobi quartic v.3a and v.3b which costs approximately 2253**M**. The same operation requires approximately 2662**M** for Weierstrass form ($a = -3$) using projective weighted (Jacobian) coordinates.

Some points representations such as the modified Hessian coordinates require extra storage for representing each point. This is certainly a disadvantage for space limited applications. However, the primary focus is on the performance in some cases where the processor bandwidth is low.

Speed implementations. This is the most difficult case in which to state a fair comparison because the optimum speeds are somewhat dependent on the choice of the scalar multiplication algorithm. For instance, Doche/Icart/Kohel-3 curves in (Doche et al. 2006) have very fast tripling formulae which can highly benefit from double base number system scalar multiplication. For double-and-add type scalar multiplication algorithms, one might expect to gain the best timing with the system which has the fastest doubling operation since point doubling is the dominant operation. However, the readdition and the mixed addition costs also play important roles in the overall timings. We can *roughly* state that the fast systems for $\mathbf{S}/\mathbf{M} = 0.8, \mathbf{D}/\mathbf{M} \approx 0$ are modified Jacobi quartics v.1, v.2a, v.2b, v.3a, v.3b, inverted Edwards v.1a, v.1b, Edwards v.2, and modified Jacobi intersection. At least, these systems can be very competitive with the Montgomery ladder which has the fixed cost of $5\mathbf{M} + 4\mathbf{S} + 1\mathbf{D}$ per scalar bit in (Montgomery 1987) and $4\mathbf{M}+5\mathbf{S}+3\mathbf{D}$ in (Castryck et al. 2008) for Montgomery curves and $3\mathbf{M}+6\mathbf{S}+3\mathbf{D}$ in (Gaudry & Lubicz 2008) for Kummer surfaces (the genus 1 case).

To make the comparison easier, we fix the *"signed 4-bit sliding windows"* scalar multiplication algorithm analyzed in (Bernstein & Lange 2007*c*). The algorithm requires 0.98 doublings, 0.17 readditions, 0.025 mixed additions and 0.0035 additions per scalar bit for 256-bit scalars. We use this analysis to report current rankings between different systems in Table 2.

With our improvements, either modified Jacobi quartic v.2b or v.3b provides the fastest timings for almost all **S**/**M** and **D**/**M** values. For example, 256-bit scalar multiplication ($\mathbf{S}/\mathbf{M} = 0.8$, $\mathbf{D}/\mathbf{M} \approx 0$) costs approximately 1970**M** for modified Jacobi quartic v.3a, v.3b. The same operation requires approximately 2399**M** for Weierstrass form ($a = -3$) using projective weighted (Jacobian) coordinates.

Side channel attacks. Targeting the embedded implementations, we fix the *"Non-adjacent form without precomputation with SPA protection"* scalar multiplication algorithm. This is almost the same as using the *"Non-adjacent form without precomputation"* method with the difference that unified addition is used for both point doubling and point addition. This strategy hides the side channel information from the attacker who needs more samplings and statistical tools for a successful attack. See Cohen et al. (2005, Section 29.1.2) as a general reference. This algorithm invokes 4/3 unified additions per scalar bit. The modified coordinates for Hessian and Jacobi intersection forms are only useful here. The $7\mathbf{M}+3\mathbf{S}+1\mathbf{D}$ unified addition of modified Jacobi quartic v.2b, v.3b is the fastest among all other unified additions. The cost estimates for various systems are depicted in Table 3.

For example, 256-bit scalar multiplication ($\mathbf{S}/\mathbf{M} = 0.8, \mathbf{D}/\mathbf{M} \approx 0$) costs approximately 3208**M** for modified Jacobi quartic v.2b, v.3b. The same operation requires approximately 5257**M** for Weierstrass form ($a = -3$) using homogenous projective coordinates.

References

Bernstein, D. J., Birkner, P., Joye, M., Lange, T. & Peters, C. (2008), Twisted Edwards curves, *in* 'AFRICACRYPT 2008', Vol. 5023 of *LNCS*, Springer, pp. 389–405.

Bernstein, D. J., Birkner, P., Lange, T. & Peters, C. (2007), Optimizing double-base elliptic-curve single-scalar multiplication, *in* 'INDOCRYPT 2007', Vol. 4859 of *LNCS*, Springer, pp. 167–182.

Bernstein, D. J. & Lange, T. (2007*a*), 'Analysis and optimization of elliptic-curve single-scalar multiplication', Cryptology ePrint Archive, Report 2007/455. `http://eprint.iacr.org/`.

Bernstein, D. J. & Lange, T. (2007*b*), 'Explicit-formulas database'. `http://www.hyperelliptic.org/EFD`.

Bernstein, D. J. & Lange, T. (2007*c*), Faster addition and doubling on elliptic curves, *in* 'ASIACRYPT 2007', Vol. 4833 of *LNCS*, Springer, pp. 29–50.

Bernstein, D. J. & Lange, T. (2007*d*), Inverted Edwards coordinates, *in* 'AAECC-17', Vol. 4851 of *LNCS*, Springer, pp. 20–27.

Billet, O. & Joye, M. (2003), The Jacobi model of an elliptic curve and side-channel analysis, *in* 'AAECC-15', Vol. 2643 of *LNCS*, Springer, pp. 34–42.

Brier, E. & Joye, M. (2002), Weierstraß elliptic curves and side-channel attacks, *in* 'PKC 2002', Vol. 2274 of *LNCS*, Springer, pp. 335–345.

Castryck, W., Galbraith, S. & Rezaeian Farashahi, R. (2008), 'Efficient arithmetic on elliptic curves using a mixed Edwards-Montgomery representation', Cryptology ePrint Archive, Report 2008/218 version 2008-06-03. `http://eprint.iacr.org/`.

Chudnovsky, D. V. & Chudnovsky, G. V. (1986), 'Sequences of numbers generated by addition in formal groups and new primality and factorization tests', *Advances in Applied Mathematics* **7**(4), 385–434.

Cohen, H., Frey, G., Avanzi, R., Doche, C., Lange, T., Nguyen, K. & Vercauteren, F., eds (2005), *Handbook of Elliptic and Hyperelliptic Curve Cryptography*, CRC Press.

Cohen, H., Miyaji, A. & Ono, T. (1998), Efficient elliptic curve exponentiation using mixed coordinates, *in* 'ASIACRYPT'98', Vol. 1514 of *LNCS*, Springer, pp. 51–65.

Doche, C., Icart, T. & Kohel, D. R. (2006), Efficient scalar multiplication by isogeny decompositions, *in* 'PKC 2006', Vol. 3958 of *LNCS*, Springer, pp. 191–206.

Duquesne, S. (2007), 'Improving the arithmetic of elliptic curves in the Jacobi model', *Information Processing Letters* **104**(3), 101–105.

Euler, L. (1761), 'De integratione aequationis differentialis $m\,dx/\sqrt{1-x^4} = n\,dy/\sqrt{1-y^4}$', *Novi Commentarii Academiae Scientiarum Petropolitanae 6* pp. 37–57. Translated from the Latin by Stacy G. Langton; On the integration of the differential equation $m\,dx/\sqrt{1-x^4} = n\,dy/\sqrt{1-y^4}$; available at `http://home.sandiego.edu/~langton/eell.pdf`.

Gaudry, P. & Lubicz, D. (2008), 'The arithmetic of characteristic 2 Kummer surfaces', Cryptology ePrint Archive, Report 2008/133 version 2008-03-25. `http://eprint.iacr.org/`.

Hisil, H., Carter, G. & Dawson, E. (2007), New formulae for efficient elliptic curve arithmetic, *in* 'INDOCRYPT 2007', Vol. 4859 of *LNCS*, Springer, pp. 138–151.

Hisil, H., Wong, K. K.-H., Carter, G. & Dawson, E. (2008), Twisted Edwards curves revisited, *in* 'ASIACRYPT 2008', Vol. 5350 of *LNCS*, Springer, pp. 326–343.

Jacobi, C. G. J. (1829), *Fundamenta nova theoriae functionum ellipticarum*, Sumtibus Fratrum Borntræger.

Joye, M. & Quisquater, J. J. (2001), Hessian elliptic curves and side-channel attacks, *in* 'CHES 2001', Vol. 2162 of *LNCS*, Springer, pp. 402–410.

Koblitz, N. (1987), 'Elliptic curve cryptosystems', *Mathematics of Computation* **48**(177), 203–209.

Liardet, P. Y. & Smart, N. P. (2001), Preventing SPA/DPA in ECC systems using the Jacobi form., *in* 'CHES 2001', Vol. 2162 of *LNCS*, Springer, pp. 391–401.

McKean, H. & Moll, V. (1927), *A Course of Modern Analysis*, Cambridge University Press.

Miller, V. S. (1986), Use of elliptic curves in cryptography, *in* 'CRYPTO'85', Vol. 218 of *LNCS*, Springer, pp. 417–426.

Monagan, M. & Pearce, R. (2006), Rational simplification modulo a polynomial ideal, *in* 'ISSAC'06', ACM, pp. 239–245.

Montgomery, P. L. (1987), 'Speeding the Pollard and elliptic curve methods of factorization', *Mathematics of Computation* **48**(177), 243–264.

Smart, N. P. (2001), The Hessian form of an elliptic curve, *in* 'CHES 2001', Vol. 2162 of *LNCS*, Springer, pp. 118–125.

A Appendix

The appendix is composed of three tables. The underlined values are the fastest timings in that column. The rows are sorted with respect to the column ($\mathbf{D} = 0, \mathbf{S} = 0.8\mathbf{M}$) in descending order. "REG" stands for the number of coordinates in each system. "DBL", "mADD", "reADD", "ADD", and "uADD" stand for the costs of doubling, mixed addition, readdition, addition and unified addition, respectively. Some forms have alternative versions due to alternative operation counts for different $\mathbf{S}/\mathbf{M}$ and $\mathbf{D}/\mathbf{M}$ values. It is possible to include more versions due to the richness of current formulae and algorithms. On the other hand, this will decrease readability of the tables. Therefore, we only provide the most significant cases. The references for the comparisons are;

- Doche/Icart/Kohel-2; all operations from (Doche et al. 2006) and (Bernstein & Lange 2007*b*). The appearance of (Bernstein & Lange 2007*b*) is to emphasize that faster algorithms are available and are obtained from this database. This is the same for other items in the list.

- Edwards; all operations for v.1a, v.1b, and doubling for v.2 from (Bernstein & Lange 2007*c*).

- Hessian; doubling for v.1, v.2 from (Hisil et al. 2007), readdition, mixed addition, and addition for v.1, addition for v.2 from (Chudnovsky & Chudnovsky 1986).

- Inverted Edwards; all operations for v.1 and doubling, readdition and addition for v.2 from (Bernstein & Lange 2007*d*).

- Jacobian ($a = -3$) and Jacobian; all operations from (Chudnovsky & Chudnovsky 1986), (Cohen et al. 1998), and (Bernstein & Lange 2007*b*).

- Jacobi intersection; doubling, addition, readdition, from (Liardet & Smart 2001) and (Bernstein & Lange 2007*b*), mixed addition from (Hisil et al. 2007).

- Jacobi quartic; doubling and addition for v.1a, v.1b from (Billet & Joye 2003), (Duquesne 2007), and (Bernstein & Lange 2007*b*). We note that the $2\mathbf{M} + 6\mathbf{S} + 2\mathbf{D}$ doubling formulae/algorithm by Hisil, Dawson and Carter reported in (Bernstein & Lange 2007*b*) cost $1\mathbf{M} + 7\mathbf{S} + 2\mathbf{D}$ if the coordinate X_3 is computed as $(X_1Z_1 + Y_1)^2 - (X_1Z_1)^2 - Y_1^2$. Jacobi quartic; readdition, mixed addition from (Billet & Joye 2003), (Duquesne 2007), and (Bernstein & Lange 2007*b*).

- Modified Jacobi quartic; doubling for v.1, v.2a, v.2b (Hisil et al. 2007) and (Bernstein & Lange 2007*b*), readdition, mixed-addition, and addition for v.1 from (Duquesne 2007) and (Bernstein & Lange 2007*b*).

- Projective ($a = -3$) and Projective; doubling, readdition, mixed addition and addition for (Chudnovsky & Chudnovsky 1986) and (Bernstein & Lange 2007*b*), unified addition from (Brier & Joye 2002) and (Bernstein & Lange 2007*b*).

The rest of the operation counts are from this paper and they are given in **bold** type in Table 1, Table 2, and Table 3.

Table 1: Point multiplication cost estimates (in $\mathbf{M}$) per scalar bit of the scalar for "Non-adjacent form without precomputation" method. The underlined values are the fastest timing estimates in that column. The rows are sorted with respect to the column ($\mathbf{D} = 0, \mathbf{S} = 0.8\mathbf{M}$) in descending order. The new operation counts are given in **bold**.

System	REG	DBL			mADD			1 DBL, 1 / 3 mADD per bit								
		M	S	D	M	S	D	D=M S=M	D=M S=0.8M	D=M S=0.67M	D=0.5M S=M	D=0.5M S=0.8M	D=0.5M S=0.67M	D=0 S=M	D=0 S=0.8M	D=0 S=0.67M
Projective	3	5	6	1	9	2	0	15.667	14.333	13.467	15.167	13.833	12.967	14.667	13.333	12.467
Projective (a=-3)	3	7	3	0	9	2	0	13.667	12.933	12.457	13.667	12.933	12.457	13.667	12.933	12.457
Jacobi-quartic v.1a	3	1	9	0	7	3	1	13.667	11.667	10.367	13.500	11.500	10.200	13.333	11.333	10.033
Hessian v.1	3	7	1	0	10	0	0	11.333	11.133	11.003	11.333	11.133	11.003	11.333	11.133	11.003
Hessian v.2	3	3	6	0	**5**	**6**	**0**	12.667	11.067	10.027	12.667	11.067	10.027	12.667	11.067	10.027
Modified Hessian	9	**3**	**6**	**0**	**5**	**6**	**0**	12.667	11.067	10.027	12.667	11.067	10.027	12.667	11.067	10.027
Jacobian	3	1	8	1	7	4	0	13.667	11.800	10.587	13.167	11.300	10.087	12.667	10.800	9.587
Jacobian (a=-3)	3	3	5	0	7	4	0	11.667	10.400	9.577	11.667	10.400	9.577	11.667	10.400	9.577
Jacobi-intersection v.1	4	3	4	0	10	2	1	11.333	10.400	9.793	11.167	10.233	9.627	11.000	10.067	9.460
Jacobi-quartic v.1b	3	1	7	2	7	3	1	13.667	12.067	11.027	12.500	10.900	9.860	11.333	9.733	8.693
Doche/Icart/Kohel-2	4	2	5	2	8	4	1	13.333	12.067	11.243	12.167	10.900	10.077	11.000	9.733	8.910
Jacobi-intersection v.2	4	**2**	**5**	**1**	**10**	**1**	**2**	12.333	11.267	10.573	11.500	10.433	9.740	10.667	9.600	8.907
Modified Jacobi-intersection	6	**2**	**5**	**1**	**10**	**1**	**2**	12.333	11.267	10.573	11.500	10.433	9.740	10.667	9.600	8.907
Edwards v.1b	3	3	4	0	6	5	1	11.000	9.867	9.130	10.833	9.700	8.963	10.667	9.533	8.797
Edwards v.1a	3	3	4	0	9	1	1	10.667	9.800	9.237	10.500	9.633	9.070	10.333	9.467	8.903
Modified Jacobi-quartic v.1	6	3	4	0	7	3	1	10.667	9.667	9.017	10.500	9.500	8.850	10.333	9.333	8.683
Inverted Edwards v.2	3	3	4	1	**9**	**0**	**0**	11.000	10.200	9.680	10.500	9.700	9.180	10.000	9.200	8.680
Edwards v.2	3	3	4	0	**9**	**0**	**0**	10.000	9.200	8.680	10.000	9.200	8.680	10.000	9.200	8.680
Inverted Edwards v.1	3	3	4	1	8	1	1	11.333	10.467	9.903	10.667	9.800	9.237	10.000	9.133	8.570
Modified Jacobi-quartic v.2a	5	3	4	0	**6**	**3**	**1**	10.333	9.333	8.683	10.167	9.167	8.517	10.000	9.000	8.350
Modified Jacobi-quartic v.2b	6	3	4	0	**6**	**3**	**1**	10.333	9.333	8.683	10.167	9.167	8.517	10.000	9.000	8.350
Modified Jacobi-quartic v.3a	5	**2**	**5**	**1**	**6**	**3**	**1**	11.333	10.133	9.353	10.667	9.467	8.687	10.000	8.800	8.020
Modified Jacobi-quartic v.3b	6	**2**	**5**	**1**	**6**	**3**	**1**	11.333	10.133	9.353	10.667	9.467	8.687	10.000	8.800	8.020

Table 2: Point multiplication cost estimates (in $\mathbf{M}$) per scalar bit of the scalar for "Signed 4-bit Sliding Windows" method with 256 bit scalars. The underlined values are the fastest timing estimates in that column. The rows are sorted with respect to the column ($\mathbf{D} = 0, \mathbf{S} = 0.8\mathbf{M}$) in descending order. The new operation counts are given in **bold**.

System	REG	DBL			reADD			mADD			ADD			0.98 DBL, 0.17 reADD, 0.025 mADD, 0.0035 ADD per bit								
		M	S	D	M	S	D	M	S	D	M	S	D	D=M S=M	D=M S=0.8M	D=M S=0.67M	D=0.5M S=M	D=0.5M S=0.8M	D=0.5M S=0.67M	D=0 S=M	D=0 S=0.8M	D=0 S=0.67M
Projective	3	5	6	1	12	2	0	9	2	0	12	2	0	14.433	13.177	12.360	13.942	12.685	11.869	13.451	12.194	11.377
Projective (a=-3)	3	7	3	0	12	2	0	9	2	0	12	2	0	12.468	11.801	11.368	12.468	11.801	11.368	12.468	11.801	11.368
Jacobi-quartic v.1a	3	1	9	0	8	3	1	7	3	1	10	3	1	12.136	10.251	9.026	12.039	10.154	8.929	11.942	10.057	8.832
Hessian v.1	3	7	1	0	12	0	0	10	0	0	12	0	0	10.140	9.943	9.816	10.140	9.943	9.816	10.140	9.943	9.816
Jacobian	3	1	8	1	10	4	0	7	4	0	11	5	0	12.475	10.748	9.624	11.984	10.256	9.133	11.493	9.765	8.642
Hessian v.2	3	3	6	0	**6**	**6**	**0**	**5**	**6**	**0**	12	0	0	11.147	9.739	8.824	11.147	9.739	8.824	11.147	9.739	8.824
Modified Hessian	9	**3**	**6**	**0**	**6**	**6**	**0**	**5**	**6**	**0**	**6**	**6**	**0**	11.147	9.735	8.817	11.147	9.735	8.817	11.147	9.735	8.817
Jacobian (a=-3)	3	3	5	0	10	4	0	7	4	0	11	5	0	10.511	9.372	8.632	10.511	9.372	8.632	10.511	9.372	8.632
Doche/Icart/Kohel-2	4	2	5	2	12	5	1	8	4	1	12	5	1	12.213	11.042	10.280	11.134	9.962	9.201	10.054	8.883	8.121
Jacobi-intersection v.1	4	3	4	0	11	2	1	10	2	1	13	2	1	9.577	8.714	8.152	9.480	8.617	8.055	9.383	8.520	7.958
Jacobi-quartic v.1b	3	1	7	2	8	3	1	7	3	1	10	3	1	12.136	10.644	9.675	11.057	9.565	8.595	9.977	8.485	7.516
Edwards v.1b	3	3	4	0	7	5	1	6	5	1	7	5	1	9.376	8.396	7.759	9.279	8.299	7.662	9.182	8.202	7.565
Jacobi-intersection v.2	4	**2**	**5**	**1**	**11**	**1**	**2**	**10**	**1**	**2**	**13**	**1**	**2**	10.560	9.539	8.875	9.874	8.853	8.189	9.189	8.168	7.504
Edwards v.1a	3	3	4	0	10	1	1	9	1	1	10	1	1	9.182	8.357	7.821	9.085	8.260	7.724	8.988	8.163	7.627
Modified Jacobi-intersection	6	**2**	**5**	**1**	**11**	**1**	**2**	**10**	**1**	**2**	**11**	**1**	**2**	10.553	9.531	8.868	9.867	8.846	8.182	9.182	8.161	7.497
Edwards v.2	3	3	4	0	**9**	**2**	**0**	**9**	**0**	**0**	**11**	**0**	**0**	8.963	8.111	7.557	8.963	8.111	7.557	8.963	8.111	7.557
Modified Jacobi-quartic v.1	6	3	4	0	8	3	1	7	3	1	8	3	1	9.182	8.280	7.693	9.085	8.183	7.596	8.988	8.085	7.499
Inverted Edwards v.2	3	3	4	1	9	1	1	**9**	**0**	**0**	9	1	1	9.946	9.126	8.593	9.370	8.550	8.017	8.794	7.974	7.441
Inverted Edwards v.1	3	3	4	1	9	1	1	8	1	1	9	1	1	9.970	9.146	8.609	9.382	8.557	8.021	8.794	7.969	7.433
Modified Jacobi-quartic v.2a	5	3	4	0	**7**	**3**	**1**	**6**	**3**	**1**	**7**	**4**	**1**	8.991	8.088	7.501	8.894	7.991	7.404	8.797	7.894	7.307
Modified Jacobi-quartic v.2b	6	3	4	0	**7**	**3**	**1**	**6**	**3**	**1**	**7**	**3**	**1**	8.988	8.085	7.499	8.891	7.988	7.402	8.794	7.891	7.305
Modified Jacobi-quartic v.3a	5	**2**	**5**	**1**	**7**	**3**	**1**	**6**	**3**	**1**	**7**	**4**	**1**	9.974	8.874	8.159	9.386	8.286	7.571	8.797	7.698	6.983
Modified Jacobi-quartic v.3b	6	**2**	**5**	**1**	**7**	**3**	**1**	**6**	**3**	**1**	**7**	**3**	**1**	9.970	8.871	8.157	9.382	8.283	7.569	8.794	7.695	6.981

Table 3: Point multiplication cost estimates (in $\mathbf{M}$) per scalar bit of the scalar for “Non-adjacent form without precomputation with SPA protection” method. The underlined values are the fastest timing estimates in that column. The rows are sorted with respect to the column ($\mathbf{D} = 0, \mathbf{S} = 0.8\mathbf{M}$) in descending order. The new operation counts are given in **bold**.

System	REG	uADD			4 / 3 uADD per bit								
		M	S	D	D=M S=M	D=M S=0.8M	D=M S=0.67M	D=0.5M S=M	D=0.5M S=0.8M	D=0.5M S=0.67M	D=0 S=M	D=0 S=0.8M	D=0 S=0.67M
Projective	3	11	6	1	24.000	22.400	21.360	23.333	21.733	20.693	22.667	21.067	20.027
Projective (a=-1)	3	13	3	0	21.333	20.533	20.013	21.333	20.533	20.013	21.333	20.533	20.013
Jacobi-intersection v.1	4	13	2	1	21.333	20.800	20.453	20.667	20.133	19.787	20.000	19.467	19.120
Jacobi-intersection v.2	4	**13**	**1**	**2**	21.333	21.067	20.893	20.000	19.733	19.560	18.667	18.400	18.227
Jacobi-quartic v.1a, v.1b	3	10	3	1	18.667	17.867	17.347	18.000	17.200	16.680	17.333	16.533	16.013
Hessian v.1, v.2	3	12	0	0	16.000	16.000	16.000	16.000	16.000	16.000	16.000	16.000	16.000
Modified Jacobi-intersection	6	**11**	**1**	**2**	18.667	18.400	18.227	17.333	17.067	16.893	16.000	15.733	15.560
Edwards v.1b	3	7	5	1	17.333	16.000	15.133	16.667	15.333	14.467	16.000	14.667	13.800
Edwards v.1a	3	10	1	1	16.000	15.733	15.560	15.333	15.067	14.893	14.667	14.400	14.227
Modified Hessian	9	**6**	**6**	**0**	16.000	14.400	13.360	16.000	14.400	13.360	16.000	14.400	13.360
Modified Jacobi-quartic v.1	6	8	3	1	16.000	15.200	14.680	15.333	14.533	14.013	14.667	13.867	13.347
Modified Jacobi-quartic v.2a, v.3a	5	**7**	**4**	**1**	16.000	14.933	14.240	15.333	14.267	13.573	14.667	13.600	12.907
Inverted Edwards v.1	3	9	1	1	14.667	14.400	14.227	14.000	13.733	13.560	13.333	13.067	12.893
Modified Jacobi-quartic v.2b, v.3b	6	**7**	**3**	**1**	14.667	13.867	13.347	14.000	13.200	12.680	13.333	12.533	12.013