Faster addition and doubling on elliptic curves

Daniel J. Bernstein[1] and Tanja Lange[2]

[1] Department of Mathematics, Statistics, and Computer Science (M/C 249)
University of Illinois at Chicago, Chicago, IL 60607–7045, USA
djb@cr.yp.to

[2] Department of Mathematics and Computer Science
Technische Universiteit Eindhoven, P.O. Box 513, 5600 MB Eindhoven, Netherlands
tanja@hyperelliptic.org

Abstract. Edwards recently introduced a new normal form for elliptic curves. Every elliptic curve over a non-binary field is birationally equivalent to a curve in Edwards form over an extension of the field, and in many cases over the original field.

This paper presents fast explicit formulas (and register allocations) for group operations on an Edwards curve. The algorithm for doubling uses only $3\mathbf{M}+4\mathbf{S}$, i.e., 3 field multiplications and 4 field squarings. If curve parameters are chosen to be small then the algorithm for mixed addition uses only $9\mathbf{M}+1\mathbf{S}$ and the algorithm for non-mixed addition uses only $10\mathbf{M}+1\mathbf{S}$. Arbitrary Edwards curves can be handled at the cost of just one extra multiplication by a curve parameter.

For comparison, the fastest algorithms known for the popular "$a_4 = -3$ Jacobian" form use $3\mathbf{M}+5\mathbf{S}$ for doubling; use $7\mathbf{M}+4\mathbf{S}$ for mixed addition; use $11\mathbf{M}+5\mathbf{S}$ for non-mixed addition; and use $10\mathbf{M}+4\mathbf{S}$ for non-mixed addition when one input has been added before.

The explicit formulas for non-mixed addition on an Edwards curve can be used for doublings at no extra cost, simplifying protection against side-channel attacks. Even better, many elliptic curves (approximately 1/4 of all isomorphism classes of elliptic curves over a non-binary finite field) are birationally equivalent — over the original field — to Edwards curves where this addition algorithm works for *all* pairs of curve points, including inverses, the neutral element, etc.

This paper contains an extensive comparison of different forms of elliptic curves and different coordinate systems for the basic group operations (doubling, mixed addition, non-mixed addition, and unified addition) as well as higher-level operations such as multi-scalar multiplication.

Keywords: elliptic curves, addition, doubling, explicit formulas, register allocation, scalar multiplication, multi-scalar multiplication, side-channel countermeasures, unified addition formulas, complete addition formulas, efficient implementation, performance evaluation

1 Introduction

The core operations in elliptic-curve cryptography are single-scalar multiplication $(m, P \mapsto mP)$, double-scalar multiplication $(m, n, P, Q \mapsto mP + nQ)$, etc. Miller, in his Crypto '85 paper introducing elliptic-curve cryptography, proposed carrying out these operations on points represented in Jacobian form: "Each point is represented by the triple (x, y, z) which corresponds to the point $(x/z^2, y/z^3)$" on a curve $y^2 = x^3 + a_4x + a_6$. See [35, page 424]. One can add two points using 16 field multiplications, specifically $11\mathbf{M}+5\mathbf{S}$, with the fastest algorithms known today; here we keep separate tallies of squarings $\mathbf{S}$ and general multiplications $\mathbf{M}$. A mixed addition — this means that one input has $z = 1$ — takes only $7\mathbf{M}+4\mathbf{S}$. A doubling takes $1\mathbf{M}+8\mathbf{S}+1\mathbf{D}$, where $\mathbf{D}$ denotes the cost of multiplying by a_4; a doubling takes $3\mathbf{M}+5\mathbf{S}$ in the special case $a_4 = -3$.

[*] Permanent ID of this document: 95616567a6ba20f575c5f25e7cebaf83. Date of this document: 2007.07.16.

Several subsequent papers analyzed the performance of other forms of elliptic curves proposed in the mathematical literature. See, e.g., [17] for the speed of several dialects of the Weierstrass form, [33] for the speed of Jacobi intersections, [27] for the speed of Hessians, and [8] for the speed of Jacobi quartics; see also [36] and [22], which introduced the Montgomery and Doche/Icart/Kohel forms and analyzed their speed. These alternate forms attracted some interest — in particular, many of them simplify protection against side-channel attacks, and the speed records in [7] for single-scalar multiplication were set with the Montgomery form — but the Jacobian form remained the overall speed leader for multi-scalar multiplication.

A new form for elliptic curves was added to the mathematical literature a few months ago: Edwards showed in [24] that all elliptic curves over number fields could be transformed to the shape $x^2 + y^2 = c^2(1 + x^2y^2)$, with $(0, c)$ as neutral element and with the surprisingly simple and symmetric addition law

$$(x_1, y_1), (x_2, y_2) \mapsto \left(\frac{x_1y_2 + y_1x_2}{c(1 + x_1x_2y_1y_2)}, \frac{y_1y_2 - x_1x_2}{c(1 - x_1x_2y_1y_2)} \right).$$

Similarly, all elliptic curves over non-binary finite fields can be transformed to Edwards form. Some elliptic curves require a field extension for the transformation, but some elliptic curves have transformations defined over the original number field or finite field.

To capture a larger class of elliptic curves over the original field, we expand the notion of Edwards form to include all curves $x^2 + y^2 = c^2(1 + dx^2y^2)$ where $cd(1 - dc^4) \neq 0$. More than 1/4 of all isomorphism classes of elliptic curves over a finite field — for example, the curve "Curve25519" previously used to set speed records for single-scalar multiplication — can be transformed to Edwards curves over the same field. See Sections 2 and 3 of this paper for further background on Edwards curves.

Our main goal in this paper is to analyze the impact of Edwards curves upon cryptographic applications. Our main conclusions are that the Edwards form (1) breaks solidly through the Jacobian speed barrier, (2) is competitive with the Montgomery form for single-scalar multiplication, and (3) is the new speed leader for multi-scalar multiplication. Specifically, we present explicit formulas (i.e., sequences of additions, subtractions, and multiplications) that

- compute an addition $(X_1 : Y_1 : Z_1), (X_2 : Y_2 : Z_2) \mapsto (X_1 : Y_1 : Z_1) + (X_2 : Y_2 : Z_2)$ using $10\mathbf{M} + 1\mathbf{S} + 1\mathbf{D}$ — here $\mathbf{D}$ is the cost of multiplying by a selectable curve parameter;
- compute a mixed addition $(X_1 : Y_1 : Z_1), (X_2 : Y_2 : 1) \mapsto (X_1 : Y_1 : Z_1) + (X_2 : Y_2 : 1)$ using $9\mathbf{M} + 1\mathbf{S} + 1\mathbf{D}$; and
- compute a doubling $(X_1 : Y_1 : Z_1) \mapsto 2(X_1 : Y_1 : Z_1)$ using $3\mathbf{M} + 4\mathbf{S}$.

See Section 4 for details of these computations; Section 5 for a comparison of these speeds to the speeds of explicit formulas for Jacobian, Hessian, etc.; Sections 6 and 7 for an analysis of the resulting speeds of single-scalar multiplication and general multi-scalar multiplication; and Section 8 for a discussion of side-channel attacks.

An Edwards curve with a unique point of order 2 has the extra feature that the addition formulas are *complete*. This means that the formulas work for *all* pairs of input points on the curve, with no exceptions for doubling, no exceptions for the neutral element, no exceptions for negatives, etc. Some previous addition formulas have been advertised as *unified* formulas that can handle generic doublings, simplifying protection against side-channel attacks; our addition formulas are faster than previous unified formulas and have the stronger property of completeness. See Sections 3, 5, and 8 for further discussion.

Acknowledgments. We thank Harold M. Edwards for his comments and encouragement, and of course for finding the Edwards addition law in the first place.

We thank Marc Joye for suggesting using the curve equation to accelerate the computation of the x-coordinate of $2P$. See Section 4.

This work has been supported in part by the European Commission through the IST Programme under Contract IST–2002–507932 ECRYPT. This work was carried out while the first author was visiting Technische Universiteit Eindhoven.

2 Transformation to Edwards form

Fix a field k of characteristic different from 2. Let E be an elliptic curve over k having a point of order 4. This section shows that some quadratic twist of E is birationally equivalent over k to an Edwards curve: specifically, a curve of the form $x^2 + y^2 = 1 + dx^2y^2$ with $d \notin \{0, 1\}$. (Perhaps this twist is E itself; perhaps not.) Section 3 shows that the Edwards addition law on the Edwards curve corresponds to the standard elliptic-curve addition law.

If E has a unique point of order 2 then some quadratic twist of E is birationally equivalent over k to an Edwards curve having non-square d. If k is finite and E has a unique point of order 2 then the twist can be removed: E is birationally equivalent over k to an Edwards curve having non-square d. Section 3 shows that the Edwards addition law is complete in this case.

All of these equivalences can be computed efficiently. The proof of Theorem 2.1 explicitly constructs d given a Weierstrass-form elliptic curve, and explicitly maps points between the Weierstrass curve and the Edwards curve.

As an example, consider the elliptic curve published in [7] for fast scalar multiplication in Montgomery form, namely the elliptic curve $v^2 = u^3 + 486662u^2 + u$ modulo $p = 2^{255} - 19$. This curve "Curve25519" is birationally equivalent over $\mathbf{Z}/p$ to the Edwards curve $x^2 + y^2 = 1 + (121665/121666)x^2y^2$. The transformation is easy: simply define $x = \sqrt{486664}u/v$ and $y = (u - 1)/(u + 1)$; note that 486664 is a square modulo p. The inverse transformation is just as easy: simply define $u = (1 + y)/(1 - y)$ and $v = x/\sqrt{486664}u$.

Every Edwards curve has a point of order 4; see Section 3. So it is natural to consider elliptic curves having points of order 4. What about elliptic curves that do not have points of order 4 — for example, the NIST curves over prime fields? Construct an extension field k' of k such that $E(k')$, the group of points of E defined over k', has an element of order 4. Then replace k by k' in Theorem 2.1 to see that some twist of E is birationally equivalent over k' to an Edwards curve defined over k'.

Theorem 2.1. *Let k be a field in which $2 \neq 0$. Let E be an elliptic curve over k such that the group $E(k)$ has an element of order 4. Then*

(1) there exists $d \in k - \{0, 1\}$ such that the curve $x^2 + y^2 = 1 + dx^2y^2$ is birationally equivalent over k to a quadratic twist of E;
(2) if $E(k)$ has a unique element of order 2 then there is a nonsquare $d \in k$ such that the curve $x^2 + y^2 = 1 + dx^2y^2$ is birationally equivalent over k to a quadratic twist of E; and
(3) if k is finite and $E(k)$ has a unique element of order 2 then there is a nonsquare $d \in k$ such that the curve $x^2 + y^2 = 1 + dx^2y^2$ is birationally equivalent over k to E.

Proof. Write E in long Weierstrass form $s^2 + a_1rs + a_3s = r^3 + a_2r^2 + a_4r + a_6$. Assume without loss of generality that $a_1 = 0$ and $a_3 = 0$; to handle the general case, define $\overline{s} = s + (a_1r + a_3)/2$.

Write P for the hypothesized point of order 4 on E. Assume without loss of generality that $2P = (0,0)$ and thus $a_6 = 0$; to handle the general case, define $\overline{r} = r - r_2$ where $2P = (r_2, s_2)$.

The elliptic curve E now has the form $s^2 = r^3 + a_2r^2 + a_4r$. Write P as (r_1, s_1). The next step is to express the curve coefficients a_2 and a_4 in terms of r_1 and s_1.

Note that $s_1 \neq 0$, as otherwise P has order 2. Consequently $r_1 \neq 0$. The equation $2P = (0,0)$ means that the tangent line to E at P passes through $(0,0)$, i.e., that $s_1 - 0 = (r_1 - 0)\lambda$ where λ is the tangent slope $(3r_1^2 + 2a_2r_1 + a_4)/2s_1$. Thus $3r_1^3 + 2a_2r_1^2 + a_4r_1 = 2s_1^2$. Also $2s_1^2 = 2r_1^3 + 2a_2r_1^2 + 2a_4r_1$ since P is on the curve. Subtract to see that $r_1^3 = a_4r_1$, i.e., $r_1^2 = a_4$. Furthermore $a_2 = (s_1^2 - r_1^3 - a_4r_1)/r_1^2 = s_1^2/r_1^2 - 2r_1$. Putting $d = 1 - 4r_1^3/s_1^2$ we obtain $a_2 = 2((1+d)/(1-d))r_1$.

Note that $d \neq 1$ since $r_1 \neq 0$. Note also that $d \neq 0$: otherwise the right hand side of E's equation would be $r^3 + a_2r^2 + a_4r = r^3 + 2r_1r^2 + r_1^2r = r(r + r_1)^2$, contradicting the hypothesis that E is elliptic. Note also that if d is a square then there is another point of order 2 in $E(k)$, namely $\bigl(r_1(\sqrt{d}+1)/(\sqrt{d}-1), 0\bigr)$.

Consider two quadratic twists of E, namely the elliptic curve E' defined by $(r_1/(1-d))s^2 = r^3 + a_2r^2 + a_4r$ and the elliptic curve E'' defined by $(dr_1/(1-d))s^2 = r^3 + a_2r^2 + a_4r$.

If k is finite and d is nonsquare then either $r_1/(1-d)$ or $dr_1/(1-d)$ is a square in k so E is isomorphic to either E' or E''.

Substitute $u = r/r_1$ and $v = s/r_1$ to see that E' is isomorphic to the elliptic curve $(1/(1-d))v^2 = u^3 + 2((1+d)/(1-d))u^2 + u$ and that E'' is isomorphic to $(d/(1-d))v^2 = u^3 + 2((1+d)/(1-d))u^2 + u$.

We now show that the curve $x^2+y^2 = 1+dx^2y^2$ is birationally equivalent to $(1/(1-d))v^2 = u^3 + 2((1+d)/(1-d))u^2 + u$, and therefore to E'. The rational map $(u,v) \mapsto (x,y)$ is defined by $x = 2u/v$ and $y = (u-1)/(u+1)$; observe that

$$\begin{aligned}
x^2 + y^2 - 1 - dx^2y^2 &= \frac{4u^2}{v^2} + \frac{(u-1)^2}{(u+1)^2} - 1 - d\frac{4u^2}{v^2}\frac{(u-1)^2}{(u+1)^2} \\
&= \frac{1}{(u+1)^2v^2}\bigl(4u^2(u+1)^2 + (u-1)^2v^2 - (u+1)^2v^2 - 4du^2(u-1)^2\bigr) \\
&= \frac{4u}{(u+1)^2v^2}\bigl(u(u+1)^2 - v^2 - du(u-1)^2\bigr) \\
&= \frac{4u}{(u+1)^2v^2}\bigl((1-d)u^3 + 2(1+d)u^2 + (1-d)u - v^2\bigr) \\
&= \frac{4(1-d)u}{(u+1)^2v^2}\left(u^3 + 2\frac{1+d}{1-d}u^2 + u - \frac{1}{1-d}v^2\right) = 0.
\end{aligned}$$

There are only finitely many exceptional points with $v(u+1) = 0$. The inverse rational map $(x,y) \mapsto (u,v)$ is defined by $u = (1+y)/(1-y)$ and $v = 2(1+y)/(1-y)x$; there are only finitely many exceptional points with $(1-y)x = 0$.

Substitute $1/d$ for d and $-u$ for u to see that $x^2 + y^2 = 1 + (1/d)x^2y^2$ is birationally equivalent to the curve $(1/(1-1/d))v^2 = (-u)^3 + 2((1+1/d)/(1-1/d))(-u)^2 + (-u)$, i.e., to $(d/(1-d))v^2 = u^3 + 2((1+d)/(1-d))u^2 + u$, and therefore to E''.

To summarize: (1) The curve $x^2 + y^2 = 1 + dx^2y^2$ is equivalent to a quadratic twist E' of E. (2) If E has a unique point of order 2 then d is a nonsquare and $x^2 + y^2 = 1 + dx^2y^2$ is equivalent to a quadratic twist E' of E. (3) If k is finite and E has a unique point of order 2 then d is a nonsquare so E is isomorphic to E' or to E''; thus E is birationally equivalent to $x^2 + y^2 = 1 + dx^2y^2$ or to $x^2 + y^2 = 1 + (1/d)x^2y^2$. $\square$

Notes on isomorphisms. If $d = \overline{d}\overline{c}^4$ then the curve $x^2 + y^2 = 1 + dx^2y^2$ is isomorphic to the curve $\overline{x}^2 + \overline{y}^2 = \overline{c}^2(1 + \overline{d}\overline{x}^2\overline{y}^2)$: simply define $\overline{x} = \overline{c}x$ and $\overline{y} = \overline{c}y$.

In particular, if k is a finite field, then at least $1/4$ of the nonzero elements of k are 4th powers, so $d/\overline{d}$ is a 4th power for at least $1/4$ of the choices of $\overline{d} \in k - \{0\}$; the smallest qualifying $\overline{d}$ is typically extremely small. But for computational purposes we do not recommend minimizing $\overline{d}$ as a general strategy: $\overline{c}$ will usually be quite large, and a small $\overline{c}$ is more valuable than a small $\overline{d}$. See Section 4.

3 The Edwards addition law

This section presents the Edwards addition law for an Edwards curve $x^2+y^2 = c^2(1+dx^2y^2)$. We show (1) that the Edwards addition law produces points on the curve, (2) that the Edwards addition law corresponds to the standard addition law on a birationally equivalent elliptic curve, and (3) that the Edwards addition law is complete when d is not a square. We postpone proofs until the end of the section.

Fix a field k of characteristic different from 2. Fix $c, d \in k$ such that $c \neq 0$, $d \neq 0$, and $dc^4 \neq 1$. Consider the *Edwards addition law*

$$(x_1, y_1), (x_2, y_2) \mapsto \left(\frac{x_1y_2 + y_1x_2}{c(1 + dx_1x_2y_1y_2)}, \frac{y_1y_2 - x_1x_2}{c(1 - dx_1x_2y_1y_2)}\right)$$

on the Edwards curve $x^2 + y^2 = c^2(1 + dx^2y^2)$ over k.

Examples: for each point $P = (x_1, y_1)$ on the curve, P is the sum of $(0, c)$ and P, so $(0, c)$ is a neutral element of the addition law; the only neutral element is $(0, c)$; $(0, c)$ is the sum of P and $-P = (-x_1, y_1)$; in particular, $(0, -c)$ has order 2; $(c, 0)$ and $(-c, 0)$ have order 4.

The next theorem states that the output of the Edwards addition law is on the curve when the output is defined, i.e., when the denominators $1 \pm dx_1x_2y_1y_2$ are nonzero.

Theorem 3.1. *Let k be a field in which $2 \neq 0$. Let c, d be nonzero elements of k with $dc^4 \neq 1$. Let x_1, y_1, x_2, y_2 be elements of k such that $x_1^2+y_1^2 = c^2(1+dx_1^2y_1^2)$ and $x_2^2+y_2^2 = c^2(1+dx_2^2y_2^2)$. Assume that $dx_1x_2y_1y_2 \notin \{-1, 1\}$. Define $x_3 = (x_1y_2 + y_1x_2)/c(1 + dx_1x_2y_1y_2)$ and $y_3 = (y_1y_2 - x_1x_2)/c(1 - dx_1x_2y_1y_2)$. Then $x_3^2 + y_3^2 = c^2(1 + dx_3^2y_3^2)$.*

The next theorem states that the output of the Edwards addition law corresponds to the output of the standard addition law on a birationally equivalent elliptic curve E. One can therefore perform group operations on E (or on any other birationally equivalent elliptic curve) by performing the corresponding group operations on the Edwards curve, at the expense of evaluating and inverting the correspondence once for each series of computations.

Theorem 3.2. *In the situation of Theorem 3.1, let $e = 1-dc^4$ and let E be the elliptic curve $(1/e)v^2 = u^3 + (4/e-2)u^2 + u$. For each $i \in \{1, 2, 3\}$ define P_i as follows: $P_i = \infty$ if $(x_i, y_i) = (0, c)$; $P_i = (0, 0)$ if $(x_i, y_i) = (0, -c)$; and $P_i = (u_i, v_i)$ if $x_i \neq 0$, where $u_i = (c + y_i)/(c - y_i)$ and $v_i = 2c(c + y_i)/(c - y_i)x_i$. Then $P_i \in E(k)$ and $P_1 + P_2 = P_3$.*

Here $P_1 + P_2$ means the sum of P_1 and P_2 in the standard addition law on $E(k)$. Note that $x_i \neq 0$ implies $y_i \neq c$.

The group operations could encounter exceptional points where the Edwards addition law is not defined. One can, in many applications, rely on randomization to avoid the exceptional points, or one can switch from the Edwards curve back to E when exceptional points occur.

The next theorem states that, when d is not a square, there are no exceptional points: the denominators in the Edwards addition law cannot be zero. In other words, when d is not a square, the Edwards addition law is *complete*: it is defined for *all* pairs of input points on the Edwards curve over k. The set $E(k)$, with the standard addition law, is isomorphic as a group to the set of points $(x_1, y_1) \in k \times k$ on the Edwards curve, with the Edwards addition law. Any sequence of group operations can be carried out by the Edwards addition law, with no risk of failure.

Theorem 3.3. *Let k be a field in which $2 \neq 0$. Let c, d, e be nonzero elements of k with $e = 1 - dc^4$. Assume that d is not a square in k. Let x_1, y_1, x_2, y_2 be elements of k such that $x_1^2+y_1^2 = c^2(1+dx_1^2y_1^2)$ and $x_2^2+y_2^2 = c^2(1+dx_2^2y_2^2)$. Then $dx_1x_2y_1y_2 \neq 1$ and $dx_1x_2y_1y_2 \neq -1$.*

Example: $d = 121665/121666$ is not a square in the field $k = \mathbf{Z}/(2^{255} - 19)$. The Edwards addition law is defined for all $(x_1, y_1), (x_2, y_2)$ on the Edwards curve $x^2 + y^2 = 1 + dx^2y^2$ over k, and corresponds to the standard addition law on "Curve25519," the elliptic curve $v^2 = u^3 + 486662u^2 + u$ over k. The point at ∞ on Curve25519 corresponds to the point $(0, 1)$ on the Edwards curve; the point $(0, 0)$ on Curve25519 corresponds to $(0, -1)$; any other point (u, v) on Curve25519 corresponds to $(\sqrt{486664}u/v, (u - 1)/(u + 1))$; a sum of points on Curve25519 corresponds to a sum of points on the Edwards curve. One can therefore perform a sequence of group operations on points of the elliptic curve $v^2 = u^3 + 486662u^2 + u$ by performing the same sequence of group operations on the corresponding points of the Edwards curve.

The reader might wonder why [10, Theorem 1] ("The smallest cardinality of a complete system of addition laws on E equals two") does not force exceptional cases in the addition law for the curve $x^2 + y^2 = c^2(1 + dx^2y^2)$. The simplest answer is that [10, Theorem 1] is concerned with exceptional cases in the algebraic closure of k, whereas we are concerned with exceptional cases in k itself.

The reader might also wonder why we ignore the two projective points $(0 : 1 : 0)$ and $(1 : 0 : 0)$ on the Edwards curve. The answer is that, although these points might at first glance appear to be defined over k, they are actually singularities of the curve, and resolving the singularities produces four points that are defined over $k(\sqrt{d})$, not over k.

Proof (of Theorem 3.1). The special case $d = 1$ is equivalent to [24, Theorem 8.1]. We could deduce the general case from the special case, but to keep this paper self-contained we instead give a direct proof.

The first ingredient in the proof is a mechanically verifiable polynomial identity. Define $T = (x_1y_2+y_1x_2)^2(1-dx_1x_2y_1y_2)^2+(y_1y_2-x_1x_2)^2(1+dx_1x_2y_1y_2)^2-d(x_1y_2+y_1x_2)^2(y_1y_2-x_1x_2)^2$. The identity says that $T = (x_1^2 + y_1^2 - (x_2^2 + y_2^2)dx_1^2y_1^2)(x_2^2 + y_2^2 - (x_1^2 + y_1^2)dx_2^2y_2^2)$.

The second ingredient is the curve equation, i.e., the hypotheses on (x_1, y_1) and (x_2, y_2). Subtract the equation $(x_2^2 + y_2^2)dx_1^2y_1^2 = c^2(1 + dx_2^2y_2^2)dx_1^2y_1^2$ from the equation $x_1^2 + y_1^2 = c^2(1 + dx_1^2y_1^2)$ to see that $x_1^2 + y_1^2 - (x_2^2 + y_2^2)dx_1^2y_1^2 = c^2(1 - d^2x_1^2x_2^2y_1^2y_2^2)$. Similarly $x_2^2 + y_2^2 - (x_1^2 + y_1^2)dx_2^2y_2^2 = c^2(1 - d^2x_1^2x_2^2y_1^2y_2^2)$. Thus $T = c^4(1 - d^2x_1^2x_2^2y_1^2y_2^2)^2$.

The third ingredient is the Edwards addition law, i.e., the definition of (x_3, y_3) in terms of x_1, x_2, y_1, y_2. We have

$$
\begin{aligned}
&x_3^2 + y_3^2 - c^2 d x_3^2 y_3^2 \\
&\quad = \frac{(x_1y_2 + y_1x_2)^2}{c^2(1 + dx_1x_2y_1y_2)^2} + \frac{(y_1y_2 - x_1x_2)^2}{c^2(1 - dx_1x_2y_1y_2)^2} - \frac{c^2d(x_1y_2 + y_1x_2)^2(y_1y_2 - x_1x_2)^2}{c^4(1 + dx_1x_2y_1y_2)^2(1 - dx_1x_2y_1y_2)^2} \\
&\quad = \frac{T}{c^2(1 + dx_1x_2y_1y_2)^2(1 - dx_1x_2y_1y_2)^2} = \frac{T}{c^2(1 - d^2x_1^2x_2^2y_1^2y_2^2)^2} = c^2.
\end{aligned}
$$

Thus $x_3^2 + y_3^2 = c^2(1 + dx_3^2y_3^2)$ as claimed. $\square$

Proof (of Theorem 3.2). First we show that each P_i is in $E(k)$. If $(x_i, y_i) = (0, c)$ then $P_i = \infty \in E(k)$. If $(x_i, y_i) = (0, -c)$ then $P_i = (0, 0) \in E(k)$. Otherwise $P_i = (u_i, v_i) \in E(k)$ by essentially the same calculations as in Theorem 2.1, omitted here.

All that remains is to show that $P_1 + P_2 = P_3$. There are several cases in the standard addition law for $E(k)$; the proof is correspondingly forced to split into several cases.

If $(x_1, y_1) = (0, c)$ then $(x_3, y_3) = (x_2, y_2)$. Now P_1 is the point at infinity and $P_2 = P_3$, so $P_1 + P_2 = \infty + P_2 = P_2 = P_3$. Similar comments apply if $(x_2, y_2) = (0, c)$. Assume from now on that $(x_1, y_1) \neq (0, c)$ and $(x_2, y_2) \neq (0, c)$.

If $(x_3, y_3) = (0, c)$ then $(x_2, y_2) = (-x_1, y_1)$. If $(x_1, y_1) = (0, -c)$ then also $(x_2, y_2) = (0, -c)$ and $P_1 = (0, 0) = P_2$; otherwise x_1, x_2 are nonzero so $u_1 = (c + y_1)/(c - y_1) = u_2$ and $v_1 = 2cu_1/x_1 = -2cu_2/x_2 = -v_2$ so $P_1 = -P_2$. In both cases $P_1 + P_2 = \infty = P_3$. Assume from now on that $(x_3, y_3) \neq (0, c)$.

If $(x_1, y_1) = (0, -c)$ then $(x_3, y_3) = (-x_2, -y_2)$. Now $(x_2, y_2) \neq (0, -c)$ (since otherwise $(x_3, y_3) = (0, c)$) and $(x_2, y_2) \neq (0, c)$ so $x_2 \neq 0$. Thus $P_1 = (0, 0)$ and $P_2 = (u_2, v_2)$ with $u_2 = (c+y_2)/(c-y_2)$ and $v_2 = 2cu_2/x_2$. The standard addition law says that $(0, 0)+(u_2, v_2) = (r_3, s_3)$ where $r_3 = (1/e)(v_2/u_2)^2 - (4/e - 2) - u_2 = 1/u_2$ and $s_3 = (v_2/u_2)(-r_3) = -v_2/u_2^2$. Furthermore $P_3 = (u_3, v_3)$ with $u_3 = (c + y_3)/(c - y_3) = (c - y_2)/(c + y_2) = 1/u_2 = r_3$ and $v_3 = 2cu_3/x_3 = -2c/u_2x_2 = -v_2/u_2^2 = s_3$. Thus $P_1 + P_2 = P_3$. Similar comments apply if $(x_2, y_2) = (0, -c)$.

Assume from now on that $x_1 \neq 0$ and $x_2 \neq 0$. Then $P_1 = (u_1, v_1)$ with $u_1 = (c+y_1)/(c-y_1)$ and $v_1 = 2cu_1/x_1$, and $P_2 = (u_2, v_2)$ with $u_2 = (c + y_2)/(c - y_2)$ and $v_2 = 2cu_2/x_2$.

If $(x_3, y_3) = (0, -c)$ then $(x_1, y_1) = (x_2, -y_2)$ so $u_1 = (c+y_1)/(c-y_1) = (c-y_2)/(c+y_2) = 1/u_2$ and $v_1 = 2cu_1/x_1 = v_2/u_2^2$. Furthermore $P_3 = (0, 0)$ so the standard addition law says as above that $-P_3 + P_2 = (0, 0) + P_2 = (1/u_2, -v_2/u_2^2) = (u_1, -v_1) = -P_1$, i.e., $P_1 + P_2 = P_3$.

Assume from now on that $x_3 \neq 0$. Then $P_3 = (u_3, v_3)$ with $u_3 = (c + y_3)/(c - y_3)$ and $v_3 = 2cu_3/x_3$.

If $P_2 = -P_1$ then $u_2 = u_1$ and $v_2 = -v_1$, so $x_2 = -x_1$ and $y_2 = c(u_2 - 1)/(u_2 + 1) = c(u_1 - 1)/(u_1 + 1) = y_1$, so $(x_3, y_3) = (0, c)$, which is already handled above. Assume from now on that $P_2 \neq -P_1$.

If $u_2 = u_1$ and $v_2 \neq -v_1$ then the standard addition law says that $(u_1, v_1) + (u_2, v_2) = (r_3, s_3)$ where $\lambda = (3u_1^2 + 2(4/e - 2)u_1 + 1)/((2/e)v_1)$, $r_3 = (1/e)\lambda^2 - (4/e - 2) - 2u_1$, and $s_3 = \lambda(u_1 - u_3) - v_1$. The following commands for the Magma computer-algebra system check that $(r_3, s_3) = (u_3, v_3)$:

```
K<c,d,x1>:=FieldOfFractions(PolynomialRing(Rationals(),3));
e:=1-d*c^4;
R<y1>:=PolynomialRing(K,1);
```

```
S:=quo<R|x1^2+y1^2-c^2*(1+d*x1^2*y1^2)>;
x2:=x1; y2:=y1;
x3:=(x1*y2+y1*x2)/(c*(1+d*x1*x2*y1*y2));
y3:=(y1*y2-x1*x2)/(c*(1-d*x1*x2*y1*y2));
u1:=(c+y1)/(c-y1); v1:=2*c*u1/x1; S!((1/e)*v1^2-u1^3-(4/e-2)*u1^2-u1);
u2:=(c+y2)/(c-y2); v2:=2*c*u2/x2; S!((1/e)*v2^2-u2^3-(4/e-2)*u2^2-u2);
u3:=(c+y3)/(c-y3); v3:=2*c*u3/x3; S!((1/e)*v3^2-u3^3-(4/e-2)*u3^2-u3);
lambda:=(3*u1^2+2*(4/e-2)*u1+1)/((2/e)*v1);
r3:=(1/e)*lambda^2-(4/e-2)-2*u1; s3:=lambda*(u1-u3)-v1;
S!(u3-r3); S!(v3-s3);
```

The only remaining case is that $u_2 \neq u_1$. The standard addition law says that $(u_1, v_1) + (u_2, v_2) = (r_3, s_3)$ where $\lambda = (v_2 - v_1)/(u_2 - u_1)$, $r_3 = (1/e)\lambda^2 - (4/e - 2) - u_1 - u_2$, and $s_3 = \lambda(u_1 - u_3) - v_1$. The following commands for the Magma computer-algebra system check that $(r_3, s_3) = (u_3, v_3)$:

```
K<c,d,x1,x2>:=FieldOfFractions(PolynomialRing(Rationals(),4));
e:=1-d*c^4;
R<y1,y2>:=PolynomialRing(K,2);
S:=quo<R|x1^2+y1^2-c^2*(1+d*x1^2*y1^2),x2^2+y2^2-c^2*(1+d*x2^2*y2^2)>;
x3:=(x1*y2+y1*x2)/(c*(1+d*x1*x2*y1*y2));
y3:=(y1*y2-x1*x2)/(c*(1-d*x1*x2*y1*y2));
u1:=(c+y1)/(c-y1); v1:=2*c*u1/x1; S!((1/e)*v1^2-u1^3-(4/e-2)*u1^2-u1);
u2:=(c+y2)/(c-y2); v2:=2*c*u2/x2; S!((1/e)*v2^2-u2^3-(4/e-2)*u2^2-u2);
u3:=(c+y3)/(c-y3); v3:=2*c*u3/x3; S!((1/e)*v3^2-u3^3-(4/e-2)*u3^2-u3);
lambda:=(v2-v1)/(u2-u1);
r3:=(1/e)*lambda^2-(4/e-2)-u1-u2; s3:=lambda*(u1-u3)-v1;
S!(u3-r3); S!(v3-s3);
```

Conclusion: $P_3 = P_1 + P_2$ in every case. □

Proof (of Theorem 3.3). Write $\epsilon = dx_1x_2y_1y_2$. Suppose that $\epsilon \in \{-1, 1\}$. Then $x_1, x_2, y_1, y_2 \neq 0$. Furthermore $dx_1^2y_1^2(x_2^2 + y_2^2) = c^2(dx_1^2y_1^2 + d^2x_1^2y_1^2x_2^2y_2^2) = c^2(dx_1^2y_1^2 + \epsilon^2) = c^2(1 + dx_1^2y_1^2) = x_1^2 + y_1^2$ so

$$\begin{aligned}(x_1 + \epsilon y_1)^2 = x_1^2 + y_1^2 + 2\epsilon x_1y_1 &= dx_1^2y_1^2(x_2^2 + y_2^2) + 2x_1y_1dx_1x_2y_1y_2 \\ &= dx_1^2y_1^2(x_2^2 + 2x_2y_2 + y_2^2) = dx_1^2y_1^2(x_2 + y_2)^2.\end{aligned}$$

If $x_2 + y_2 \neq 0$ then $d = ((x_1 + \epsilon y_1)/x_1y_1(x_2 + y_2))^2$ so d is a square, contradiction. Similarly, if $x_2 - y_2 \neq 0$ then $d = ((x_1 - \epsilon y_1)/x_1y_1(x_2 - y_2))^2$ so d is a square, contradiction. If both $x_2 + y_2$ and $x_2 - y_2$ are 0 then $x_2 = 0$ and $y_2 = 0$, contradiction. □

4 Efficient group operations in Edwards form

This section presents fast explicit formulas and register allocations for doubling, mixed addition, etc. on Edwards curves with arbitrary parameters c, d.

As usual we count the number of operations in the underlying field. We keep separate tallies of the number of general multiplications (each costing **M**), squarings (each costing

S), multiplications by c (each costing **C**), multiplications by d (each costing **D**), and additions/subtractions (each costing **a**). The costs $\mathbf{M}, \mathbf{S}, \mathbf{C}, \mathbf{D}, \mathbf{a}$ depend on the choice of platform, on the choice of finite field, and on the choice of c and d.

Every Edwards curve can easily be transformed to an isomorphic Edwards curve over the same field having $c = 1$ and thus $\mathbf{C} = 0$; see "Notes on isomorphisms" in Section 2. In subsequent sections we assume that $c = 1$. However, we can imagine applications in which $c \neq 1$ (for example, a curve with a fairly small c and with $d = 1$ could have smaller $\mathbf{C} + \mathbf{D}$ than an isomorphic curve with $\overline{c} = 1$ and $\overline{d} = c^4$), so we allow arbitrary (c, d) in our explicit formulas.

Addition. To avoid the inversions in the original addition formulas

$$(x_1, y_1), (x_2, y_2) \mapsto \left(\frac{x_1 y_2 + y_1 x_2}{c(1 + d x_1 x_2 y_1 y_2)}, \frac{y_1 y_2 - x_1 x_2}{c(1 - d x_1 x_2 y_1 y_2)} \right),$$

we homogenize the curve equation to $(X^2 + Y^2)Z^2 = c^2(Z^4 + dX^2Y^2)$. A point $(X_1 : Y_1 : Z_1)$ satisfying $(X_1^2 + Y_1^2)Z_1^2 = c^2(Z_1^4 + dX_1^2Y_1^2)$ and $Z_1 \neq 0$ corresponds to the affine point $(X_1/Z_1, Y_1/Z_1)$. The neutral element is $(0 : c : 1)$, and the inverse of $(X_1 : Y_1 : Z_1)$ is $(-X_1 : Y_1 : Z_1)$.

The following formulas, given $(X_1 : Y_1 : Z_1)$ and $(X_2 : Y_2 : Z_2)$, compute the sum $(X_3 : Y_3 : Z_3) = (X_1 : Y_1 : Z_1) + (X_2 : Y_2 : Z_2)$:

$$\begin{aligned}
&A = Z_1 \cdot Z_2;\ B = A^2;\ C = X_1 \cdot X_2;\ D = Y_1 \cdot Y_2;\\
&E = d \cdot C \cdot D;\ F = B - E;\ G = B + E;\\
&X_3 = A \cdot F \cdot ((X_1 + Y_1) \cdot (X_2 + Y_2) - C - D);\\
&Y_3 = A \cdot G \cdot (D - C);\ Z_3 = c \cdot F \cdot G.
\end{aligned}$$

One readily counts $10\mathbf{M} + 1\mathbf{S} + 1\mathbf{C} + 1\mathbf{D} + 7\mathbf{a}$. We have saved operations here by rewriting $x_1 y_2 + x_2 y_1$ as $(x_1 + y_1)(x_2 + y_2) - x_1 x_2 - y_1 y_2$ and by exploiting common subexpressions.

The following specific sequence of operations starts with registers R_1, R_2, R_3 containing X_1, Y_1, Z_1 and registers R_4, R_5, R_6 containing X_2, Y_2, Z_2, uses just two temporary registers R_7, R_8 and constants c, d, ends with registers R_1, R_2, R_3 containing X_3, Y_3, Z_3 and untouched registers R_4, R_5, R_6 containing X_2, Y_2, Z_2, and uses $10\mathbf{M} + 1\mathbf{S} + 1\mathbf{C} + 1\mathbf{D} + 7\mathbf{a}$:

$$\begin{aligned}
&R_3 \leftarrow R_3 \cdot R_6;\ R_7 \leftarrow R_1 + R_2;\ R_8 \leftarrow R_4 + R_5;\ R_1 \leftarrow R_1 \cdot R_4;\\
&R_2 \leftarrow R_2 \cdot R_5;\ R_7 \leftarrow R_7 \cdot R_8;\ R_7 \leftarrow R_7 - R_1;\ R_7 \leftarrow R_7 - R_2;\\
&R_7 \leftarrow R_7 \cdot R_3;\ R_8 \leftarrow R_1 \cdot R_2;\ R_8 \leftarrow d \cdot R_8;\ R_2 \leftarrow R_2 - R_1;\\
&R_2 \leftarrow R_2 \cdot R_3;\ R_3 \leftarrow R_3^2;\ R_1 \leftarrow R_3 - R_8;\ R_3 \leftarrow R_3 + R_8;\\
&R_2 \leftarrow R_2 \cdot R_3;\ R_3 \leftarrow R_3 \cdot R_1;\ R_1 \leftarrow R_1 \cdot R_7;\ R_3 \leftarrow c \cdot R_3.
\end{aligned}$$

We emphasize that these formulas work whether or not $(X_1 : Y_1 : Z_1) = (X_2 : Y_2 : Z_2)$. There is no need to go to extra effort to unify the addition formulas with separate doubling formulas; the addition formulas are already unified. See Section 3 for further discussion of the scope of validity of the addition formulas. In particular, we emphasize that the addition law works for *all* pairs of input points if d is not a square.

As an alternative, one can obtain $A(B - E)$ and $A(B + E)$ and $(B - E)(B + E)$ as linear combinations of $A^2, B^2, E^2, (A + B)^2, (A + E)^2$. This change replaces $10\mathbf{M} + 1\mathbf{S}$ by $7\mathbf{M} + 5\mathbf{S}$, presumably saving time on platforms where $\mathbf{S}/\mathbf{M} < 0.75$. Note that $\mathbf{S}/\mathbf{M} \approx 0.67$ in [7].

Mixed addition. “Mixed addition” refers to the case that Z_2 is known to be 1. In this case the multiplication $A = Z_1 \cdot Z_2$ can be eliminated, reducing the total costs to $9\mathbf{M}+1\mathbf{S}+1\mathbf{C}+1\mathbf{D}+7\mathbf{a}$:

$$\begin{aligned} B &= Z_1^2;\ C = X_1 \cdot X_2;\ D = Y_1 \cdot Y_2;\\ E &= d \cdot C \cdot D;\ F = B - E;\ G = B + E;\\ X_3 &= Z_1 \cdot F \cdot ((X_1 + Y_1) \cdot (X_2 + Y_2) - C - D);\\ Y_3 &= Z_1 \cdot G \cdot (D - C);\ Z_3 = c \cdot F \cdot G. \end{aligned}$$

Doubling. “Doubling” refers to the case that $(X_1 : Y_1 : Z_1)$ and $(X_2 : Y_2 : Z_2)$ are known to be equal. In this case the following formulas (with $2H$ computed as $H + H$) use only $3\mathbf{M} + 4\mathbf{S} + 3\mathbf{C} + 6\mathbf{a}$:

$$\begin{aligned} &B = (X_1 + Y_1)^2;\ C = X_1^2;\ D = Y_1^2;\ E = C + D;\ H = (c \cdot Z_1)^2;\\ &J = E - 2H;\ X_3 = c \cdot (B - E) \cdot J;\ Y_3 = c \cdot E \cdot (C - D);\ Z_3 = E \cdot J. \end{aligned}$$

We have saved operations here by rewriting $c(1+dx_1^2y_1^2)$ as $(x_1^2+y_1^2)/c$ using the curve equation, by rewriting $c(1 - dx_1^2y_1^2)$ as $(2c^2 - (x_1^2 + y_1^2))/c$, by rewriting $2x_1y_1$ as $(x_1 + y_1)^2 - x_1^2 - y_1^2$, and by exploiting common subexpressions. Thanks to Marc Joye for suggesting rewriting $c(1 + dx_1^2y_1^2)$ as $(x_1^2 + y_1^2)/c$.

The following specific sequence of operations, starting with registers R_1, R_2, R_3 containing X_1, Y_1, Z_1, changes registers R_1, R_2, R_3 to contain X_3, Y_3, Z_3, using $3\mathbf{M} + 4\mathbf{S} + 3\mathbf{C} + 6\mathbf{a}$ and using just two temporary registers R_4, R_5:

$$\begin{aligned} &R_4 \leftarrow R_1 + R_2;\ R_3 \leftarrow c \cdot R_3;\ R_1 \leftarrow R_1^2;\ R_2 \leftarrow R_2^2;\\ &R_3 \leftarrow R_3^2;\ R_4 \leftarrow R_4^2;\ R_3 \leftarrow R_3 + R_3;\ R_5 \leftarrow R_1 + R_2;\\ &R_2 \leftarrow R_1 - R_2;\ R_4 \leftarrow R_4 - R_5;\ R_3 \leftarrow R_5 - R_3;\ R_1 \leftarrow R_3 \cdot R_4;\\ &R_3 \leftarrow R_3 \cdot R_5;\ R_2 \leftarrow R_2 \cdot R_5;\ R_1 \leftarrow c \cdot R_1;\ R_2 \leftarrow c \cdot R_2. \end{aligned}$$

The following alternate sequence of operations uses one more addition, totalling $3\mathbf{M} + 4\mathbf{S} + 3\mathbf{C} + 7\mathbf{a}$, but uses just one additional register R_4:

$$\begin{aligned} &R_3 \leftarrow c \cdot R_3;\ R_4 \leftarrow R_1^2;\ R_1 \leftarrow R_1 + R_2;\ R_1 \leftarrow R_1^2;\ R_2 \leftarrow R_2^2;\\ &R_3 \leftarrow R_3^2;\ R_3 \leftarrow R_3 + R_3;\ R_4 \leftarrow R_2 + R_4;\ R_2 \leftarrow R_2 + R_2;\\ &R_2 \leftarrow R_4 - R_2;\ R_1 \leftarrow R_1 - R_4;\ R_2 \leftarrow R_2 \cdot R_4;\ R_3 \leftarrow R_4 - R_3;\\ &R_1 \leftarrow R_1 \cdot R_3;\ R_3 \leftarrow R_3 \cdot R_4;\ R_1 \leftarrow c \cdot R_1;\ R_2 \leftarrow c \cdot R_2. \end{aligned}$$

Another option is to scale $(X_3 : Y_3 : Z_3)$ to $(X_3/c : Y_3/c : Z_3/c)$, replacing two multiplications by c with one multiplication by $1/c$; typically $1/c$ can be precomputed. Of course, all three multiplications by c can be skipped if $c = 1$.

5 Comparison to previous addition speeds

This section compares the speeds of the algorithms in Section 4 to the speeds of previous algorithms for elliptic-curve doubling, elliptic-curve mixed addition, etc. The next three sections perform similar comparisons for higher-level elliptic-curve operations relevant to various cryptographic applications.

Level of detail of the comparison. We follow most of the literature in ignoring the costs of additions, subtractions, and multiplications by small constants. We recognize that these costs (and the costs of non-arithmetic operations) can be quite noticeable in practice, and we plan a more detailed cost evaluation of the Edwards form along the lines of [7], but for this paper we ignore the costs.

Consider, for example, the usual doubling algorithm for Jacobian coordinates in the case $a_4 = -3$: there are 4 squarings, 4 general multiplications, 5 additions and subtractions, and 5 multiplications by the small constants $2, 3, 4, 8, 8$. We summarize these costs as $4\mathbf{M} + 4\mathbf{S}$.

Some algorithms involve multiplications by curve parameters, such as the parameter d in Edwards curves. Some applications can take advantage of multiplying by a constant d, and some applications can choose curves where d is small, but other applications cannot. To cover both situations we separately tally the cost $\mathbf{D}$ of multiplying by a curve parameter; the reader can substitute $\mathbf{D} = 0$, $\mathbf{D} = \mathbf{M}$, or anything in between.

Each of our tables includes a column "$(1, 1)$" that substitutes $(\mathbf{S}, \mathbf{D}) \approx (\mathbf{M}, \mathbf{M})$, a column "$(0.8, 0.5)$" that substitutes $(\mathbf{S}, \mathbf{D}) \approx (0.8\mathbf{M}, 0.5\mathbf{M})$, and a column "$(0.8, 0)$" that substitutes $(\mathbf{S}, \mathbf{D}) \approx (0.8\mathbf{M}, 0\mathbf{M})$. We sort each table using the standard, but debatable, approximations $(\mathbf{S}, \mathbf{D}) \approx (0.8\mathbf{M}, 0\mathbf{M})$. We do not claim that these approximations are valid for most applications. The order of entries in our tables can easily be affected by small changes in the $\mathbf{S}/\mathbf{M}$ ratio, the $\mathbf{D}/\mathbf{M}$ ratio, etc.

Algorithms in the literature. We have built an "Explicit-Formulas Database" containing, in computer-readable format, various algorithms for operations on elliptic curves. The database is publicly available at `http://www.hyperelliptic.org/EFD`. It currently consists of 96 scripts for the Magma computer-algebra system checking the correctness of algorithms for elliptic curves in the following forms:

- **Projective**: A point (x, y) on an elliptic curve $y^2 = x^3 + ax + b$, with neutral element at infinity, is represented as $(X : Y : Z)$ satisfying $Y^2Z = X^3 + aXZ^2 + bZ^3$. Here $(X : Y : Z) = (\lambda X : \lambda Y : \lambda Z)$ for all nonzero λ.
- **Jacobian**: A point (x, y) on an elliptic curve $y^2 = x^3 + ax + b$, with neutral element at infinity, is represented as $(X : Y : Z)$ satisfying $Y^2 = X^3 + aXZ^4 + bZ^6$. Here $(X : Y : Z) = (\lambda^2 X : \lambda^3 Y : \lambda Z)$ for all nonzero λ.
- **Jacobi quartic** (with leading and trailing coefficients 1): A point (x, y) on an elliptic curve $y^2 = x^4 + 2ax^2 + 1$, with neutral element $(0, 1)$, is represented as $(X : Y : Z)$ satisfying $Y^2 = X^4 + 2aX^2Z^2 + Z^4$. Here $(X : Y : Z) = (\lambda X : \lambda^2 Y : \lambda Z)$ for all nonzero λ.
- **Jacobi intersection**: A point (s, c, d) on an elliptic curve $s^2 + c^2 = 1$, $as^2 + d^2 = 1$, with neutral element $(0, 1, 1)$, is represented as $(S : C : D : Z)$ satisfying $S^2 + C^2 = Z^2$, $aS^2 + D^2 = Z^2$. Here $(S : C : D : Z) = (\lambda S : \lambda C : \lambda D : \lambda Z)$ for all nonzero λ.
- **Hessian**: A point (x, y) on an elliptic curve $x^3 + y^3 + 1 = 3axy$, with neutral element at infinity, is represented as $(X : Y : Z)$ satisfying $X^3 + Y^3 + Z^3 = 3aXYZ$. Here $(X : Y : Z) = (\lambda X : \lambda Y : \lambda Z)$ for all nonzero λ.
- **Doche/Icart/Kohel**: A point (x, y) on an elliptic curve $y^2 = x^3 + ax^2 + 16ax$, with neutral element at infinity, is represented as $(X : Y : Z : Z^2)$ satisfying $Y^2 = ZX^3 + aZ^2X^2 + 16aZ^3X$. Here $(X : Y : Z : Z^2) = (\lambda X : \lambda^2 Y : \lambda Z : \lambda^2 Z^2)$ for all nonzero λ.
- **Edwards** (with $c = 1$): A point (x, y) on an elliptic curve $x^2 + y^2 = 1 + dx^2y^2$, with neutral element $(0, 1)$, is represented as $(X : Y : Z)$ satisfying $(X^2 + Y^2)Z^2 = Z^4 + dX^2Y^2$. Here $(X : Y : Z) = (\lambda X : \lambda Y : \lambda Z)$ for all nonzero λ.

We copied formulas from several sources in the literature; see [23] for an overview. One particularly noteworthy source is the 1986 paper [15] by Chudnovsky and Chudnovsky, containing formulas and operation counts for several forms of elliptic curves: projective, Jacobian, Jacobi quartic, Jacobi intersection, and Hessian. Liardet and Smart in [33] presented faster algorithms for Jacobi intersections. Billet and Joye in [8] presented faster algorithms for Jacobi quartics. Joye and Quisquater in [27] pointed out that the Hessian addition formulas (dating back to Sylvester) could also be used for doublings after a permutation of input coordinates, providing a weak form of unification: specifically, $2(X_1 : Y_1 : Z_1) = (Z_1 : X_1 : Y_1) + (Y_1 : Z_1 : X_1)$. Brier and Joye in [12] presented unified addition formulas for projective (and affine) coordinates; see also [11]. Of course, we also include our own algorithms for Edwards curves.

Chudnovsky and Chudnovsky also pointed out, in the case of Jacobian coordinates, that *readdition* of a point is less expensive than the first addition. The addition formulas for $(X_1 : Y_1 : Z_1) + (X_2 : Y_2 : Z_2)$ use $1\mathbf{M} + 1\mathbf{S}$ to compute Z_2^2 and Z_2^3; by caching Z_2^2 and Z_2^3 one can save $1\mathbf{M} + 1\mathbf{S}$ in computing any $(X' : Y' : Z') + (X_2 : Y_2 : Z_2)$. We comment that similar savings are possible for Jacobi intersections and Jacobi quartics.

(Rather than distinguishing readditions from initial additions, Chudnovsky and Chudnovsky reported speeds for addition and doubling of points represented as $(X : Y : Z : Z^2 : Z^3)$. But this representation is wasteful, as pointed out by Cohen, Miyaji, and Ono in [17]: if $(X_1 : Y_1 : Z_1)$ is used only for a doubling and not for a general addition then there is no need to compute Z_1^3. Sometimes coordinates $(X : Y : Z : Z^2 : Z^3)$ are called "Chudnovsky coordinates" or "Chudnovsky-Jacobian coordinates," and computing Z^2 and Z^3 only when they are needed is called "mixing Chudnovsky coordinates with Jacobian coordinates." We prefer to describe the same speedup using the simpler concept of readditions.)

Our operation counts for previous systems are often better than the operation counts reported in the literature. One reason is that a multiplication can often be replaced with a squaring, saving $\mathbf{M} - \mathbf{S}$. For example, as pointed out in [5, pages 16–17], Jacobian doubling with $a = -3$ uses $3\mathbf{M} + 5\mathbf{S}$ rather than the usual $4\mathbf{M} + 4\mathbf{S}$. As another example, Doche/Icart/Kohel doubling uses $2\mathbf{M} + 5\mathbf{S} + 2\mathbf{D}$ rather than $3\mathbf{M} + 4\mathbf{S} + 2\mathbf{D}$. The Explicit-Formulas Database contains full justification for each of our operation counts.

Comparison charts. The following table reports speeds for addition of two points:

System	ADD	$(1,1)$	$(0.8,0.5)$	$(0.8,0)$
Doche/Icart/Kohel	$12\mathbf{M} + 5\mathbf{S} + 1\mathbf{D}$	$18\mathbf{M}$	$16.5\mathbf{M}$	$16\mathbf{M}$
Jacobian	$11\mathbf{M} + 5\mathbf{S}$	$16\mathbf{M}$	$15\mathbf{M}$	$15\mathbf{M}$
Jacobi intersection	$13\mathbf{M} + 2\mathbf{S} + 1\mathbf{D}$	$16\mathbf{M}$	$15.1\mathbf{M}$	$14.6\mathbf{M}$
Projective	$12\mathbf{M} + 2\mathbf{S}$	$14\mathbf{M}$	$13.6\mathbf{M}$	$13.6\mathbf{M}$
Jacobi quartic	$10\mathbf{M} + 3\mathbf{S} + 1\mathbf{D}$	$14\mathbf{M}$	$12.9\mathbf{M}$	$12.4\mathbf{M}$
Hessian	$12\mathbf{M}$	$12\mathbf{M}$	$12\mathbf{M}$	$12\mathbf{M}$
Edwards	$10\mathbf{M} + 1\mathbf{S} + 1\mathbf{D}$	$12\mathbf{M}$	$11.3\mathbf{M}$	$10.8\mathbf{M}$

Readdition of a point already used in an addition:

System	reADD	$(1,1)$	$(0.8,0.5)$	$(0.8,0)$
Doche/Icart/Kohel	$12\mathbf{M}+5\mathbf{S}+1\mathbf{D}$	$18\mathbf{M}$	$16.5\mathbf{M}$	$16\mathbf{M}$
Projective	$12\mathbf{M}+2\mathbf{S}$	$14\mathbf{M}$	$13.6\mathbf{M}$	$13.6\mathbf{M}$
Jacobian	$10\mathbf{M}+4\mathbf{S}$	$14\mathbf{M}$	$13.2\mathbf{M}$	$13.2\mathbf{M}$
Jacobi intersection	$11\mathbf{M}+2\mathbf{S}+1\mathbf{D}$	$14\mathbf{M}$	$13.1\mathbf{M}$	$12.6\mathbf{M}$
Hessian	$12\mathbf{M}$	$12\mathbf{M}$	$12\mathbf{M}$	$12\mathbf{M}$
Jacobi quartic	$9\mathbf{M}+3\mathbf{S}+1\mathbf{D}$	$13\mathbf{M}$	$11.9\mathbf{M}$	$11.4\mathbf{M}$
Edwards	$10\mathbf{M}+1\mathbf{S}+1\mathbf{D}$	$12\mathbf{M}$	$11.3\mathbf{M}$	$10.8\mathbf{M}$

Mixed addition (i.e., addition assuming that $Z_2 = 1$):

System	mADD	$(1,1)$	$(0.8,0.5)$	$(0.8,0)$
Jacobi intersection	$11\mathbf{M}+2\mathbf{S}+1\mathbf{D}$	$14\mathbf{M}$	$13.1\mathbf{M}$	$12.6\mathbf{M}$
Doche/Icart/Kohel	$8\mathbf{M}+4\mathbf{S}+1\mathbf{D}$	$13\mathbf{M}$	$11.7\mathbf{M}$	$11.2\mathbf{M}$
Projective	$9\mathbf{M}+2\mathbf{S}$	$11\mathbf{M}$	$10.6\mathbf{M}$	$10.6\mathbf{M}$
Jacobi quartic	$8\mathbf{M}+3\mathbf{S}+1\mathbf{D}$	$12\mathbf{M}$	$10.9\mathbf{M}$	$10.4\mathbf{M}$
Jacobian	$7\mathbf{M}+4\mathbf{S}$	$11\mathbf{M}$	$10.2\mathbf{M}$	$10.2\mathbf{M}$
Hessian	$10\mathbf{M}$	$10\mathbf{M}$	$10\mathbf{M}$	$10\mathbf{M}$
Edwards	$9\mathbf{M}+1\mathbf{S}+1\mathbf{D}$	$11\mathbf{M}$	$10.3\mathbf{M}$	$9.8\mathbf{M}$

Doubling:

System	DBL	$(1,1)$	$(0.8,0.5)$	$(0.8,0)$
Projective	$5\mathbf{M}+6\mathbf{S}+1\mathbf{D}$	$12\mathbf{M}$	$10.3\mathbf{M}$	$9.8\mathbf{M}$
Projective if $a=-3$	$7\mathbf{M}+3\mathbf{S}$	$10\mathbf{M}$	$9.4\mathbf{M}$	$9.4\mathbf{M}$
Hessian	$6\mathbf{M}+3\mathbf{S}$	$9\mathbf{M}$	$8.4\mathbf{M}$	$8.4\mathbf{M}$
Jacobi quartic	$1\mathbf{M}+9\mathbf{S}+1\mathbf{D}$	$11\mathbf{M}$	$8.7\mathbf{M}$	$8.2\mathbf{M}$
Jacobian	$1\mathbf{M}+8\mathbf{S}+1\mathbf{D}$	$10\mathbf{M}$	$7.9\mathbf{M}$	$7.4\mathbf{M}$
Jacobian if $a=-3$	$3\mathbf{M}+5\mathbf{S}$	$8\mathbf{M}$	$7\mathbf{M}$	$7\mathbf{M}$
Jacobi intersection	$3\mathbf{M}+4\mathbf{S}$	$7\mathbf{M}$	$6.2\mathbf{M}$	$6.2\mathbf{M}$
Edwards	$3\mathbf{M}+4\mathbf{S}$	$7\mathbf{M}$	$6.2\mathbf{M}$	$6.2\mathbf{M}$
Doche/Icart/Kohel	$2\mathbf{M}+5\mathbf{S}+2\mathbf{D}$	$9\mathbf{M}$	$7\mathbf{M}$	$6\mathbf{M}$

Unified addition:

System	UNI	$(1,1)$	$(0.8,0.5)$	$(0.8,0)$
Projective	$11\mathbf{M}+6\mathbf{S}+1\mathbf{D}$	$18\mathbf{M}$	$16.3\mathbf{M}$	$15.8\mathbf{M}$
Projective if $a=-1$	$13\mathbf{M}+3\mathbf{S}$	$16\mathbf{M}$	$15.4\mathbf{M}$	$15.4\mathbf{M}$
Jacobi intersection	$13\mathbf{M}+2\mathbf{S}+1\mathbf{D}$	$16\mathbf{M}$	$15.1\mathbf{M}$	$14.6\mathbf{M}$
Jacobi quartic	$10\mathbf{M}+3\mathbf{S}+1\mathbf{D}$	$14\mathbf{M}$	$12.9\mathbf{M}$	$12.4\mathbf{M}$
Hessian	$12\mathbf{M}$	$12\mathbf{M}$	$12\mathbf{M}$	$12\mathbf{M}$
Edwards	$10\mathbf{M}+1\mathbf{S}+1\mathbf{D}$	$12\mathbf{M}$	$11.3\mathbf{M}$	$10.8\mathbf{M}$

Most of the addition formulas in this last table are *strongly unified*: they work without change for doublings. The Hessian addition algorithm is an exception: it works for doublings only after a permutation of input coordinates. As mentioned earlier, the addition algorithm for Edwards curves with non-square d has the stronger feature of being *complete*: it works without change for *all* inputs.

6 Single-scalar variable-point multiplication

This section compares Edwards curves to previous curve forms for single-scalar variable-point multiplication: computing nP given an integer n and a curve point P. This is one of the critical computations in elliptic-curve cryptography; for example, if n is a secret key and P is another user's public key then nP is a Diffie-Hellman secret shared between the two users. The next section considers variations of the same problem: fixed points P (allowing precomputation of, e.g., $2^{128}P$), more scalars and points, etc.

See [2] and [21] for surveys of the classic algorithms for scalar multiplication. We focus on "signed sliding window" algorithms, specifically with "window width 1" (also known as "non-adjacent form" or "NAF") or "window width 4." We also discuss the "Montgomery ladder."

We make the standard assumption that the input point P has $Z = 1$. All additions of P can thus be computed as mixed additions. By scaling other points to have $Z = 1$ one can create more mixed additions at the expense of extra field inversions; for the sake of simplicity we ignore this option in our comparison.

The NAF algorithm, for an average b-bit scalar n, uses approximately b doublings and approximately $(1/3)b$ mixed additions. So we tally the cost of 1 doubling and 1/3 mixed additions:

System	1 DBL, 1/3 mADD	$(1,1)$	$(0.8,0.5)$	$(0.8,0)$
Projective	$8\mathbf{M}+6.67\mathbf{S}+1\mathbf{D}$	$15.7\mathbf{M}$	$13.8\mathbf{M}$	$13.3\mathbf{M}$
Projective if $a=-3$	$10\mathbf{M}+3.67\mathbf{S}$	$13.7\mathbf{M}$	$12.9\mathbf{M}$	$12.9\mathbf{M}$
Hessian	$9.33\mathbf{M}+3\mathbf{S}$	$12.3\mathbf{M}$	$11.7\mathbf{M}$	$11.7\mathbf{M}$
Jacobi quartic	$3.67\mathbf{M}+10\mathbf{S}+1.33\mathbf{D}$	$15\mathbf{M}$	$12.3\mathbf{M}$	$11.7\mathbf{M}$
Jacobian	$3.33\mathbf{M}+9.33\mathbf{S}+1\mathbf{D}$	$13.7\mathbf{M}$	$11.3\mathbf{M}$	$10.8\mathbf{M}$
Jacobian if $a=-3$	$5.33\mathbf{M}+6.33\mathbf{S}$	$11.7\mathbf{M}$	$10.4\mathbf{M}$	$10.4\mathbf{M}$
Jacobi intersection	$6.67\mathbf{M}+4.67\mathbf{S}+0.333\mathbf{D}$	$11.7\mathbf{M}$	$10.6\mathbf{M}$	$10.4\mathbf{M}$
Doche/Icart/Kohel	$4.67\mathbf{M}+6.33\mathbf{S}+2.33\mathbf{D}$	$13.3\mathbf{M}$	$10.9\mathbf{M}$	$9.73\mathbf{M}$
Edwards	$6\mathbf{M}+4.33\mathbf{S}+0.333\mathbf{D}$	$10.7\mathbf{M}$	$9.63\mathbf{M}$	$9.47\mathbf{M}$

The "signed width-4 sliding windows" algorithm involves, on average, approximately $b-4.5$ doublings, $7b/48+5.2$ readditions, $b/48+0.9$ mixed additions, and 0.9 non-mixed additions; e.g., approximately 251.5 doublings, 42.5 readditions, 6.3 mixed additions, and 0.9 non-mixed additions for $b = 256$. (Different variants of the algorithm have slightly different costs; we chose one variant and measured it for 10000 uniform random 256-bit integers n.) So we tally the cost of $251.5/256 \approx 0.98$ doublings, $42.5/256 \approx 0.17$ readditions, $6.3/256 \approx 0.025$ mixed additions, and $0.9/256 \approx 0.0035$ non-mixed additions:

System	0.98 DBL, 0.17 reADD, etc.	$(1,1)$	$(0.8,0.5)$	$(0.8,0)$
Projective	$7.17\mathbf{M}+6.28\mathbf{S}+0.982\mathbf{D}$	$14.4\mathbf{M}$	$12.7\mathbf{M}$	$12.2\mathbf{M}$
Projective if $a=-3$	$9.13\mathbf{M}+3.34\mathbf{S}$	$12.5\mathbf{M}$	$11.8\mathbf{M}$	$11.8\mathbf{M}$
Hessian	$8.18\mathbf{M}+2.95\mathbf{S}$	$11.1\mathbf{M}$	$10.5\mathbf{M}$	$10.5\mathbf{M}$
Jacobi quartic	$2.71\mathbf{M}+9.42\mathbf{S}+1.18\mathbf{D}$	$13.3\mathbf{M}$	$10.8\mathbf{M}$	$10.2\mathbf{M}$
Jacobian	$2.85\mathbf{M}+8.64\mathbf{S}+0.982\mathbf{D}$	$12.5\mathbf{M}$	$10.3\mathbf{M}$	$9.77\mathbf{M}$
Jacobian if $a=-3$	$4.82\mathbf{M}+5.69\mathbf{S}$	$10.5\mathbf{M}$	$9.37\mathbf{M}$	$9.37\mathbf{M}$
Doche/Icart/Kohel	$4.2\mathbf{M}+5.86\mathbf{S}+2.16\mathbf{D}$	$12.2\mathbf{M}$	$9.96\mathbf{M}$	$8.88\mathbf{M}$
Jacobi intersection	$5.09\mathbf{M}+4.32\mathbf{S}+0.194\mathbf{D}$	$9.6\mathbf{M}$	$8.64\mathbf{M}$	$8.54\mathbf{M}$
Edwards	$4.86\mathbf{M}+4.12\mathbf{S}+0.194\mathbf{D}$	$9.18\mathbf{M}$	$8.26\mathbf{M}$	$8.16\mathbf{M}$

Another approach to high-speed single-scalar multiplication is Montgomery's algorithm in [36] for x-coordinate operations on curves in Montgomery form $y^2 = x^3 + ax^2 + x$. This algorithm does not support fast addition $P, Q \mapsto P + Q$, does not support arbitrary addition chains, and does not fit into our previous tables; but it does support fast "differential addition" $P - Q, P, Q \mapsto P + Q$, and therefore fast computation of "differential addition-subtraction chains."

In particular, the "Montgomery ladder" uses $5\mathbf{M} + 4\mathbf{S} + 1\mathbf{D}$ per bit of n to compute $P \mapsto nP$. For comparison, the NAF algorithm for Edwards curves with our formulas takes $6\mathbf{M} + 4.33\mathbf{S} + 0.333\mathbf{D}$ per bit of n, clearly slower than $5\mathbf{M} + 4\mathbf{S} + 1\mathbf{D}$ per bit. But signed width-4 sliding windows take only $4.86\mathbf{M} + 4.12\mathbf{S} + 0.194\mathbf{D}$ per bit for $b = 256$, saving $0.14\mathbf{M} - 0.12\mathbf{S} + 0.806\mathbf{D}$ per bit. Note that Edwards form is less sensitive to a large $\mathbf{D}$ than Montgomery form. Larger b's favor larger window widths, reducing the number of additions per bit and making Edwards curves even more attractive.

7 Multiple scalars, fixed points, etc.

General multi-scalar multiplication means computing $\sum n_i P_i$ given integers n_i and curve points P_i. Specific tasks are obtained by specifying the number of points, by specifying which points are known in advance, by specifying which integers are known in advance, etc. See generally [2] and [21].

We focus on four specific algorithms: the popular "joint sparse form" ("JSF") algorithm for computing $n_1P_1 + n_2P_2$, given b-bit integers n_1, n_2 and curve points P_1, P_2; the accelerated ECDSA verification algorithm in [1, page 9]; batch verification of elliptic-curve signatures, using the "Small Exponents Test" from [4, Section 3.3] and the multi-scalar multiplication algorithm that de Rooij in [19, Section 4] credits to Bos and Coster; and computation of nP for a fixed point P, using a standard "comb" table containing 90 precomputed multiples of P, essentially $2^{\{0,1,2,3,4,5\}b/6}(\{0,1\}P + \{0,1\}2^{b/24}P + \{0,1\}2^{2b/24}P + \{0,1\}2^{3b/24}P)$, normalized to have $Z = 1$.

The JSF algorithm uses about b doublings, about $(1/4)b$ mixed additions (for average n_1, n_2), and about $(1/4)b$ readditions. So we tally the cost of 1 doubling, 1/4 mixed additions, and 1/4 readditions:

System	1 DBL, 1/4 mADD, 1/4 reADD	$(1,1)$	$(0.8,0.5)$	$(0.8,0)$
Projective	$10.2\mathbf{M} + 7\mathbf{S} + 1\mathbf{D}$	$18.2\mathbf{M}$	$16.4\mathbf{M}$	$15.8\mathbf{M}$
Projective if $a = -3$	$12.2\mathbf{M} + 4\mathbf{S}$	$16.2\mathbf{M}$	$15.4\mathbf{M}$	$15.4\mathbf{M}$
Hessian	$11.5\mathbf{M} + 3\mathbf{S}$	$14.5\mathbf{M}$	$13.9\mathbf{M}$	$13.9\mathbf{M}$
Jacobi quartic	$5.25\mathbf{M} + 10.5\mathbf{S} + 1.5\mathbf{D}$	$17.2\mathbf{M}$	$14.4\mathbf{M}$	$13.7\mathbf{M}$
Jacobian	$5.25\mathbf{M} + 10\mathbf{S} + 1\mathbf{D}$	$16.2\mathbf{M}$	$13.8\mathbf{M}$	$13.2\mathbf{M}$
Jacobian if $a = -3$	$7.25\mathbf{M} + 7\mathbf{S}$	$14.2\mathbf{M}$	$12.8\mathbf{M}$	$12.8\mathbf{M}$
Doche/Icart/Kohel	$7\mathbf{M} + 7.25\mathbf{S} + 2.5\mathbf{D}$	$16.8\mathbf{M}$	$14.1\mathbf{M}$	$12.8\mathbf{M}$
Jacobi intersection	$8.5\mathbf{M} + 5\mathbf{S} + 0.5\mathbf{D}$	$14\mathbf{M}$	$12.8\mathbf{M}$	$12.5\mathbf{M}$
Edwards	$7.75\mathbf{M} + 4.5\mathbf{S} + 0.5\mathbf{D}$	$12.8\mathbf{M}$	$11.6\mathbf{M}$	$11.3\mathbf{M}$

The accelerated ECDSA verification algorithm uses about $(1/3)b$ doublings, about $(1/4)b$ mixed additions, and about $(1/4)b$ readditions. So we tally the cost of 1/3 doublings, 1/4 mixed additions, and 1/4 readditions:

System	1/3 DBL, 1/4 mADD, 1/4 reADD	$(1, 1)$	$(0.8, 0.5)$	$(0.8, 0)$
Projective	$6.92\mathbf{M} + 3\mathbf{S} + 0.333\mathbf{D}$	$10.2\mathbf{M}$	$9.48\mathbf{M}$	$9.32\mathbf{M}$
Projective if $a = -3$	$7.58\mathbf{M} + 2\mathbf{S}$	$9.58\mathbf{M}$	$9.18\mathbf{M}$	$9.18\mathbf{M}$
Doche/Icart/Kohel	$5.67\mathbf{M} + 3.92\mathbf{S} + 1.17\mathbf{D}$	$10.7\mathbf{M}$	$9.38\mathbf{M}$	$8.8\mathbf{M}$
Jacobi intersection	$6.5\mathbf{M} + 2.33\mathbf{S} + 0.5\mathbf{D}$	$9.33\mathbf{M}$	$8.62\mathbf{M}$	$8.37\mathbf{M}$
Jacobian	$4.58\mathbf{M} + 4.67\mathbf{S} + 0.333\mathbf{D}$	$9.58\mathbf{M}$	$8.48\mathbf{M}$	$8.32\mathbf{M}$
Hessian	$7.5\mathbf{M} + 1\mathbf{S}$	$8.5\mathbf{M}$	$8.3\mathbf{M}$	$8.3\mathbf{M}$
Jacobian if $a = -3$	$5.25\mathbf{M} + 3.67\mathbf{S}$	$8.92\mathbf{M}$	$8.18\mathbf{M}$	$8.18\mathbf{M}$
Jacobi quartic	$4.58\mathbf{M} + 4.5\mathbf{S} + 0.833\mathbf{D}$	$9.92\mathbf{M}$	$8.6\mathbf{M}$	$8.18\mathbf{M}$
Edwards	$5.75\mathbf{M} + 1.83\mathbf{S} + 0.5\mathbf{D}$	$8.08\mathbf{M}$	$7.47\mathbf{M}$	$7.22\mathbf{M}$

The batch-verification algorithm is not as well known as it should be, so we summarize it here for one variant of the ElGamal signature system. Fix a hash function H and a base point B on an elliptic curve over a 256-bit field. Define (R, s) as a signature of a message m under a public key K if R, K are curve points, s is a 256-bit integer, and $sB = H(R, m)R + K$. The batch-verification algorithm is given (e.g.) 100 alleged signatures (R_i, s_i) of 100 messages m_i under 100 keys K_i. The algorithm checks the equations $s_iB = H(R_i, m_i)R_i + K_i$ by choosing random 128-bit integers v_i and checking that the combination $(\sum_i v_is_i)B - \sum_i v_iH(R_im_i)R_i - \sum_i v_iK_i$ is zero. Computing this combination — a 201-scalar multiplication with 101 256-bit scalars and 100 128-bit scalars — takes about $0.8 \cdot 256$ mixed additions and about $24.4 \cdot 256$ readditions with the Bos-Coster algorithm. So we tally the cost of 0.8 mixed additions and 24.4 readditions:

System	0.8 mADD, 24.4 reADD	$(1, 1)$	$(0.8, 0.5)$	$(0.8, 0)$
Doche/Icart/Kohel	$299\mathbf{M} + 125\mathbf{S} + 25.2\mathbf{D}$	$450\mathbf{M}$	$412\mathbf{M}$	$399\mathbf{M}$
Jacobian	$274\mathbf{M} + 125\mathbf{S}$	$399\mathbf{M}$	$374\mathbf{M}$	$374\mathbf{M}$
Jacobi intersection	$326\mathbf{M} + 50.4\mathbf{S} + 25.2\mathbf{D}$	$402\mathbf{M}$	$379\mathbf{M}$	$366\mathbf{M}$
Projective	$300\mathbf{M} + 50.4\mathbf{S}$	$350\mathbf{M}$	$340\mathbf{M}$	$340\mathbf{M}$
Jacobi quartic	$250\mathbf{M} + 75.6\mathbf{S} + 25.2\mathbf{D}$	$351\mathbf{M}$	$323\mathbf{M}$	$311\mathbf{M}$
Hessian	$301\mathbf{M}$	$301\mathbf{M}$	$301\mathbf{M}$	$301\mathbf{M}$
Edwards	$251\mathbf{M} + 25.2\mathbf{S} + 25.2\mathbf{D}$	$302\mathbf{M}$	$284\mathbf{M}$	$271\mathbf{M}$

The 90-point-comb algorithm computes a b-bit fixed-point single-scalar multiplication as a 24-scalar multiplication with about $b/24$ doublings and about $15b/64 = 5.625(b/24)$ mixed additions. So we tally the cost of 1/24 doublings and 15/64 mixed additions:

System	1/24 DBL, 15/64 mADD	$(1, 1)$	$(0.8, 0.5)$	$(0.8, 0)$
Jacobi intersection	$2.7\mathbf{M} + 0.635\mathbf{S} + 0.234\mathbf{D}$	$3.57\mathbf{M}$	$3.33\mathbf{M}$	$3.21\mathbf{M}$
Projective	$2.32\mathbf{M} + 0.719\mathbf{S} + 0.0417\mathbf{D}$	$3.08\mathbf{M}$	$2.91\mathbf{M}$	$2.89\mathbf{M}$
Projective if $a = -3$	$2.4\mathbf{M} + 0.594\mathbf{S}$	$2.99\mathbf{M}$	$2.88\mathbf{M}$	$2.88\mathbf{M}$
Doche/Icart/Kohel	$1.96\mathbf{M} + 1.15\mathbf{S} + 0.318\mathbf{D}$	$3.42\mathbf{M}$	$3.03\mathbf{M}$	$2.88\mathbf{M}$
Jacobi quartic	$1.92\mathbf{M} + 1.08\mathbf{S} + 0.276\mathbf{D}$	$3.27\mathbf{M}$	$2.92\mathbf{M}$	$2.78\mathbf{M}$
Jacobian	$1.68\mathbf{M} + 1.27\mathbf{S} + 0.0417\mathbf{D}$	$2.99\mathbf{M}$	$2.72\mathbf{M}$	$2.7\mathbf{M}$
Hessian	$2.59\mathbf{M} + 0.125\mathbf{S}$	$2.72\mathbf{M}$	$2.69\mathbf{M}$	$2.69\mathbf{M}$
Jacobian if $a = -3$	$1.77\mathbf{M} + 1.15\mathbf{S}$	$2.91\mathbf{M}$	$2.68\mathbf{M}$	$2.68\mathbf{M}$
Edwards	$2.23\mathbf{M} + 0.401\mathbf{S} + 0.234\mathbf{D}$	$2.87\mathbf{M}$	$2.67\mathbf{M}$	$2.56\mathbf{M}$

Montgomery's x-coordinate algorithm in [36] can also be used for multi-scalar multiplication, but does not seem to provide competitive performance as the number of scalars increases, despite recent differential-addition-chain improvements in [6] and [13].

8 Countermeasures against side-channel attacks

The scalar-multiplication algorithms discussed in Sections 6 and 7 are often unacceptable for cryptographic hardware and embedded systems. Many secret bits of the integers n_i are leaked, through the pattern of doublings and mixed additions and non-mixed additions, to side-channel attacks such as simple power analysis. See generally [26] and [32].

One response is to use a fixed pattern of doublings, mixed additions, etc., independent of the integers n_i. Another response is to hide the pattern of doublings, mixed additions, etc. Some of these responses still leak the Hamming weight in the single-scalar case, and the total number of operations in the general case, but this information can be shielded at low cost in other ways. Of course, at a lower level, field operations must be individually shielded. In particular, an operation counted as **M** must be carried out by a multiplication unit whose time, power consumption, etc. do not depend on the inputs. Even if the inputs happen to be the same, and even if a faster squaring unit is available, the multiplication must not be carried out by the squaring unit. An operation counted as **S** can be carried out by a faster squaring unit whose time, power consumption, etc. do not depend on the input.

We focus on four specific side-channel countermeasures: non-sliding windows with digits $\{1, 2, 3, 4, 5, 6, 7, 8\}$; signed width-4 sliding windows with unified addition-or-doubling formulas; width-4 sliding windows with atomic blocks; and the Montgomery ladder. For concreteness we consider two examples of primitives: first single-scalar multiplication and then triple-scalar multiplication. Extra scalars produce extra additions, reducing the importance of doublings, as in Section 7; in particular, extra scalars make unified formulas more attractive.

We also discuss differential attacks at the end of the section.

Single-scalar multiplication. Non-sliding windows with digits $\{1, 2, 3, 4, 5, 6, 7, 8\}$ use, on average, approximately $b - 1.9$ doublings and $b/3 + 6$ readditions for single-scalar multiplication: e.g., 254.1 doublings and 91.4 readditions for $b = 256$. So we tally the cost of $254.1/256 \approx 0.99$ doublings and $91.4/256 \approx 0.36$ readditions:

System	0.99 DBL, 0.36 reADD	$(1,1)$	$(0.8, 0.5)$	$(0.8, 0)$
Projective	$9.27\mathbf{M} + 6.66\mathbf{S} + 0.99\mathbf{D}$	$16.9\mathbf{M}$	$15.1\mathbf{M}$	$14.6\mathbf{M}$
Projective if $a = -3$	$11.2\mathbf{M} + 3.69\mathbf{S}$	$14.9\mathbf{M}$	$14.2\mathbf{M}$	$14.2\mathbf{M}$
Hessian	$10.3\mathbf{M} + 2.97\mathbf{S}$	$13.2\mathbf{M}$	$12.6\mathbf{M}$	$12.6\mathbf{M}$
Jacobi quartic	$4.23\mathbf{M} + 9.99\mathbf{S} + 1.35\mathbf{D}$	$15.6\mathbf{M}$	$12.9\mathbf{M}$	$12.2\mathbf{M}$
Jacobian	$4.59\mathbf{M} + 9.36\mathbf{S} + 0.99\mathbf{D}$	$14.9\mathbf{M}$	$12.6\mathbf{M}$	$12.1\mathbf{M}$
Doche/Icart/Kohel	$6.3\mathbf{M} + 6.75\mathbf{S} + 2.34\mathbf{D}$	$15.4\mathbf{M}$	$12.9\mathbf{M}$	$11.7\mathbf{M}$
Jacobian if $a = -3$	$6.57\mathbf{M} + 6.39\mathbf{S}$	$13\mathbf{M}$	$11.7\mathbf{M}$	$11.7\mathbf{M}$
Jacobi intersection	$6.93\mathbf{M} + 4.68\mathbf{S} + 0.36\mathbf{D}$	$12\mathbf{M}$	$10.9\mathbf{M}$	$10.7\mathbf{M}$
Edwards	$6.57\mathbf{M} + 4.32\mathbf{S} + 0.36\mathbf{D}$	$11.2\mathbf{M}$	$10.2\mathbf{M}$	$10\mathbf{M}$

Signed width-4 sliding windows with unified addition-or-doubling formulas use, on average, $7b/6 + 2.5$ unified operations for single-scalar multiplication: e.g., 301.2 unified operations for

$b = 256$. So we tally the cost of $301.2/256 \approx 1.18$ unified operations:

System	1.18 UNI	$(1,1)$	$(0.8,0.5)$	$(0.8,0)$
Projective	$13\mathbf{M} + 7.08\mathbf{S} + 1.18\mathbf{D}$	$21.2\mathbf{M}$	$19.2\mathbf{M}$	$18.6\mathbf{M}$
Projective if $a = -1$	$15.3\mathbf{M} + 3.54\mathbf{S}$	$18.9\mathbf{M}$	$18.2\mathbf{M}$	$18.2\mathbf{M}$
Jacobi intersection	$15.3\mathbf{M} + 2.36\mathbf{S} + 1.18\mathbf{D}$	$18.9\mathbf{M}$	$17.8\mathbf{M}$	$17.2\mathbf{M}$
Jacobi quartic	$11.8\mathbf{M} + 3.54\mathbf{S} + 1.18\mathbf{D}$	$16.5\mathbf{M}$	$15.2\mathbf{M}$	$14.6\mathbf{M}$
Hessian	$14.2\mathbf{M}$	$14.2\mathbf{M}$	$14.2\mathbf{M}$	$14.2\mathbf{M}$
Edwards	$11.8\mathbf{M} + 1.18\mathbf{S} + 1.18\mathbf{D}$	$14.2\mathbf{M}$	$13.3\mathbf{M}$	$12.7\mathbf{M}$

Next we consider signed width-4 sliding windows with atomic blocks. Chevallier-Mames, Ciet, and Joye in [14] presented Jacobian-coordinate formulas using 10 atomic blocks for doubling and 16 atomic blocks for addition. Each block costs $1\mathbf{M}$ and consists of one field multiplication, one field addition, one field negation, and another field addition; many of the additions and negations are dummy operations. Barbosa and Page in [3] presented automatic tools that turn arbitrary explicit formulas using $m\mathbf{M} + s\mathbf{S}$ into formulas using $m + s$ atomic blocks, each consisting of one field multiplication and some number of field additions and negations, thus costing $1\mathbf{M}$. So we tally the cost of 0.98 doublings, 0.17 readditions, 0.025 mixed additions, and 0.0035 non-mixed additions, as in Section 6, except that we insist on $\mathbf{S} = \mathbf{M}$:

System	0.98 DBL, 0.17 reADD, etc., $\mathbf{S} = \mathbf{M}$	$(1,1)$	$(1,0)$
Projective	$13.5\mathbf{M} + 0.982\mathbf{D}$	$14.4\mathbf{M}$	$13.5\mathbf{M}$
Projective if $a = -3$	$12.5\mathbf{M}$	$12.5\mathbf{M}$	$12.5\mathbf{M}$
Jacobi quartic	$12.1\mathbf{M} + 1.18\mathbf{D}$	$13.3\mathbf{M}$	$12.1\mathbf{M}$
Jacobian	$11.5\mathbf{M} + 0.982\mathbf{D}$	$12.5\mathbf{M}$	$11.5\mathbf{M}$
Hessian	$11.1\mathbf{M}$	$11.1\mathbf{M}$	$11.1\mathbf{M}$
Jacobian if $a = -3$	$10.5\mathbf{M}$	$10.5\mathbf{M}$	$10.5\mathbf{M}$
Doche/Icart/Kohel	$10.1\mathbf{M} + 2.16\mathbf{D}$	$12.2\mathbf{M}$	$10.1\mathbf{M}$
Jacobi intersection	$9.41\mathbf{M} + 0.194\mathbf{D}$	$9.6\mathbf{M}$	$9.41\mathbf{M}$
Edwards	$8.99\mathbf{M} + 0.194\mathbf{D}$	$9.18\mathbf{M}$	$8.99\mathbf{M}$

The Montgomery ladder for single-scalar multiplication naturally uses a fixed double-add pattern costing only $5\mathbf{M} + 4\mathbf{S} + 1\mathbf{D}$ per bit. This combination of side-channel resistance and high speed has already attracted interest; see, e.g., [12, Section 4], [28], and [7].

We comment that, in some situations, the dummy operations in atomic blocks can be detected by fault attacks. Non-sliding windows (with nonzero digits), unified formulas, and the Montgomery ladder have the virtue of avoiding dummy operations.

Triple-scalar multiplication. Non-sliding windows with digits $\{1, 2, 3, 4, 5, 6, 7, 8\}$ use approximately 0.99 doublings and 1.08 readditions per bit for triple-scalar multiplication:

System	0.99 DBL, 1.08 reADD	$(1, 1)$	$(0.8, 0.5)$	$(0.8, 0)$
Projective	$17.9\mathbf{M} + 8.1\mathbf{S} + 0.99\mathbf{D}$	$27\mathbf{M}$	$24.9\mathbf{M}$	$24.4\mathbf{M}$
Projective if $a = -3$	$19.9\mathbf{M} + 5.13\mathbf{S}$	$25\mathbf{M}$	$24\mathbf{M}$	$24\mathbf{M}$
Doche/Icart/Kohel	$14.9\mathbf{M} + 10.3\mathbf{S} + 3.06\mathbf{D}$	$28.4\mathbf{M}$	$24.8\mathbf{M}$	$23.2\mathbf{M}$
Jacobian	$11.8\mathbf{M} + 12.2\mathbf{S} + 0.99\mathbf{D}$	$25\mathbf{M}$	$22.1\mathbf{M}$	$21.6\mathbf{M}$
Hessian	$18.9\mathbf{M} + 2.97\mathbf{S}$	$21.9\mathbf{M}$	$21.3\mathbf{M}$	$21.3\mathbf{M}$
Jacobian if $a = -3$	$13.8\mathbf{M} + 9.27\mathbf{S}$	$23\mathbf{M}$	$21.2\mathbf{M}$	$21.2\mathbf{M}$
Jacobi quartic	$10.7\mathbf{M} + 12.2\mathbf{S} + 2.07\mathbf{D}$	$24.9\mathbf{M}$	$21.5\mathbf{M}$	$20.4\mathbf{M}$
Jacobi intersection	$14.9\mathbf{M} + 6.12\mathbf{S} + 1.08\mathbf{D}$	$22.1\mathbf{M}$	$20.3\mathbf{M}$	$19.7\mathbf{M}$
Edwards	$13.8\mathbf{M} + 5.04\mathbf{S} + 1.08\mathbf{D}$	$19.9\mathbf{M}$	$18.3\mathbf{M}$	$17.8\mathbf{M}$

Signed width-4 sliding windows with unified addition-or-doubling formulas use approximately 1.54 unified operations per bit:

System	1.54 UNI	$(1, 1)$	$(0.8, 0.5)$	$(0.8, 0)$
Projective	$16.9\mathbf{M} + 9.24\mathbf{S} + 1.54\mathbf{D}$	$27.7\mathbf{M}$	$25.1\mathbf{M}$	$24.3\mathbf{M}$
Projective if $a = -1$	$20\mathbf{M} + 4.62\mathbf{S}$	$24.6\mathbf{M}$	$23.7\mathbf{M}$	$23.7\mathbf{M}$
Jacobi intersection	$20\mathbf{M} + 3.08\mathbf{S} + 1.54\mathbf{D}$	$24.6\mathbf{M}$	$23.3\mathbf{M}$	$22.5\mathbf{M}$
Jacobi quartic	$15.4\mathbf{M} + 4.62\mathbf{S} + 1.54\mathbf{D}$	$21.6\mathbf{M}$	$19.9\mathbf{M}$	$19.1\mathbf{M}$
Hessian	$18.5\mathbf{M}$	$18.5\mathbf{M}$	$18.5\mathbf{M}$	$18.5\mathbf{M}$
Edwards	$15.4\mathbf{M} + 1.54\mathbf{S} + 1.54\mathbf{D}$	$18.5\mathbf{M}$	$17.4\mathbf{M}$	$16.6\mathbf{M}$

Signed width-4 sliding windows with atomic blocks use approximately 0.98 doublings and 0.56 readditions per bit:

System	0.98 DBL, 0.56 reADD, $\mathbf{S} = \mathbf{M}$	$(1, 1)$	$(1, 0)$
Projective	$18.6\mathbf{M} + 0.98\mathbf{D}$	$19.6\mathbf{M}$	$18.6\mathbf{M}$
Projective if $a = -3$	$17.6\mathbf{M}$	$17.6\mathbf{M}$	$17.6\mathbf{M}$
Jacobian	$16.7\mathbf{M} + 0.98\mathbf{D}$	$17.6\mathbf{M}$	$16.7\mathbf{M}$
Jacobi quartic	$16.5\mathbf{M} + 1.54\mathbf{D}$	$18.1\mathbf{M}$	$16.5\mathbf{M}$
Doche/Icart/Kohel	$16.4\mathbf{M} + 2.52\mathbf{D}$	$18.9\mathbf{M}$	$16.4\mathbf{M}$
Jacobian if $a = -3$	$15.7\mathbf{M}$	$15.7\mathbf{M}$	$15.7\mathbf{M}$
Hessian	$15.5\mathbf{M}$	$15.5\mathbf{M}$	$15.5\mathbf{M}$
Jacobi intersection	$14.1\mathbf{M} + 0.56\mathbf{D}$	$14.7\mathbf{M}$	$14.1\mathbf{M}$
Edwards	$13\mathbf{M} + 0.56\mathbf{D}$	$13.6\mathbf{M}$	$13\mathbf{M}$

The Montgomery ladder can be generalized to a multi-scalar multiplication method using a fixed pattern of doublings and additions, as discussed in [6] and [13], but the performance of the generalization degrades rapidly as the number of scalars increases, as mentioned in Section 7.

Countermeasures against differential and correlation side-channel attacks. Curves in Edwards form are compatible with countermeasures against differential and correlation side-channel attacks:

- Randomized representations of scalars as addition-subtraction chains; see, e.g., [40] and [33, Section 4]. Our point representation supports arbitrary additions and subtractions.

- Randomized scalars; see, e.g., [18, Section 5.1].
- Randomized coordinates; see, e.g., [18, Section 5.3]. Our point representation is redundant and can be scaled freely: $(X_1 : Y_1 : Z_1) = (\lambda X_1 : \lambda Y_1 : \lambda Z_1)$ for any $\lambda \neq 0$.
- Randomized points, for example computing nP as $n(P + Q) - nQ$; see, e.g., [18, Section 5.2]. Our point representation supports arbitrary additions and subtractions.
- Randomized curves; see, e.g., [32, Section 29.2]. Using the generalized addition law involving c and d one can easily transfer the computation to an isomorphic curve with $\bar{c}$ and $\bar{d}$ satisfying $dc^4 = \bar{d}\bar{c}^4$. As another example, one can perform computations on a 3-isogenous curve.

We suggest using a combination of these countermeasures. In particular, point randomization or scalar randomization appears to be vital to counteract Goubin-type attacks.

References

1. Adrian Antipa, Daniel R. L. Brown, Robert P. Gallant, Robert J. Lambert, René Struik, Scott A. Vanstone, *Accelerated verification of ECDSA signatures*, in [41] (2006), 307–318. MR 2007d:94044. URL: `http://www.cacr.math.uwaterloo.ca/techreports/2005/tech_reports2005.html`. Citations in this document: §7.
2. Roberto M. Avanzi, *The complexity of certain multi-exponentiation techniques in cryptography*, Journal of Cryptology **18** (2005), 357–373. MR 2007f:94027. URL: `http://eprint.iacr.org/2002/154`. Citations in this document: §6, §7.
3. Manuel Barbosa, Daniel Page, *On the automatic construction of indistinguishable operations* (2005). URL: `http://eprint.iacr.org/2005/174`. Citations in this document: §8.
4. Mihir Bellare, Juan A. Garay, Tal Rabin, *Batch verification with applications to cryptography and checking*, in [34] (1998), 170–191. MR 99h:94043. Citations in this document: §7.
5. Daniel J. Bernstein, *A software implementation of NIST P-224* (2001). URL: `http://cr.yp.to/talks.html#2001.10.29`. Citations in this document: §5.
6. Daniel J. Bernstein, *Differential addition chains* (2006). URL: `http://cr.yp.to/papers.html#diffchain`. Citations in this document: §7, §8.
7. Daniel J. Bernstein, *Curve25519: new Diffie-Hellman speed records*, in [43] (2006), 207–228. URL: `http://cr.yp.to/papers.html#curve25519`. Citations in this document: §1, §2, §4, §5, §8.
8. Olivier Billet, Marc Joye, *The Jacobi model of an elliptic curve and side-channel analysis*, in [25] (2003), 34–42. MR 2005c:94045. URL: `http://eprint.iacr.org/2002/125`. Citations in this document: §1, §5.
9. Ian F. Blake, Gadiel Seroussi, Nigel P. Smart (editors), *Advances in elliptic curve cryptography*, London Mathematical Society Lecture Note Series, 317, Cambridge University Press, 2005. ISBN 0–521–60415–X. MR 2007g:94001. See [26].
10. Wieb Bosma, Hendrik W. Lenstra, Jr., *Complete systems of two addition laws for elliptic curves*, Journal of Number Theory **53** (1995), 229–240. MR 96f:11079. Citations in this document: §3, §3.
11. Éric Brier, Isabelle Déchène, Marc Joye, *Unified point addition formulae for elliptic curve cryptosystems*, in [38] (2004), 247–256. Citations in this document: §5.
12. Éric Brier, Marc Joye, *Weierstrass elliptic curves and side-channel attacks*, in [37] (2002), 335–345. URL: `http://www.geocities.com/MarcJoye/publications.html`. Citations in this document: §5, §8.
13. Daniel R. L. Brown, *Multi-dimensional Montgomery ladders for elliptic curves* (2006). URL: `http://eprint.iacr.org/2006/220`. Citations in this document: §7, §8.
14. Benoît Chevallier-Mames, Mathieu Ciet, Marc Joye, *Low-cost solutions for preventing simple side-channel analysis: side-channel atomicity*, IEEE Transactions on Computers **53** (2004), 760–768. URL: `http://bcm.crypto.free.fr/pdf/CCJ04.pdf`. Citations in this document: §8.
15. David V. Chudnovsky, Gregory V. Chudnovsky, *Sequences of numbers generated by addition in formal groups and new primality and factorization tests*, Advances in Applied Mathematics **7** (1986), 385–434. MR 88h:11094. Citations in this document: §5.
16. Henri Cohen, Gerhard Frey (editors), *Handbook of elliptic and hyperelliptic curve cryptography*, CRC Press, 2005. ISBN 1–58488–518–1. MR 2007f:14020. See [21], [23], [32].
17. Henri Cohen, Atsuko Miyaji, Takatoshi Ono, *Efficient elliptic curve exponentiation using mixed coordinates*, in [39] (1998), 51–65. MR 1726152. URL: `http://www.math.u-bordeaux.fr/~cohen/asiacrypt98.dvi`. Citations in this document: §1, §5.

18. Jean-Sébastien Coron, *Resistance against differential power analysis for elliptic curve cryptosystems*, in [31] (1999), 292–302. Citations in this document: §8, §8, §8.
19. Peter de Rooij, *Efficient exponentiation using precomputation and vector addition chains*, in [20] (1995), 389–399. MR 1479665. Citations in this document: §7.
20. Alfredo De Santis (editor), *Advances in cryptology: EUROCRYPT '94*, Lecture Notes in Computer Science, 950, Springer, Berlin, 1995. ISBN 3–540–60176–7. MR 98h:94001. See [19].
21. Christophe Doche, *Exponentiation*, in [16] (2005), 145–168. MR 2162725. Citations in this document: §6, §7.
22. Christophe Doche, Thomas Icart, David R. Kohel, *Efficient scalar multiplication by isogeny decompositions*, in [43] (2006), 191–206. Citations in this document: §1.
23. Christophe Doche, Tanja Lange, *Arithmetic of elliptic curves*, in [16] (2005), 267–302. MR 2162729. Citations in this document: §5.
24. Harold M. Edwards, Jr., *A normal form for elliptic curves*, Bulletin of the American Mathematical Society **44** (2007), 393–422. URL: `http://www.ams.org/bull/2007-44-03/S0273-0979-07-01153-6/home.html`. Citations in this document: §1, §3.
25. Marc Fossorier, Tom Hoeholdt, Alain Poli (editors), *Applied algebra, algebraic algorithms and error-correcting codes*, Lecture Notes in Computer Science, 2643, Springer, 2003. ISBN 3–540–40111–3. MR 2004j:94001. See [8].
26. Marc Joye, *Defences against side-channel analysis*, in [9] (2005), 87–100. Citations in this document: §8.
27. Marc Joye, Jean-Jacques Quisquater, *Hessian elliptic curves and side-channel attacks*, in [30] (2001), 402–410. MR 2003k:94032. URL: `http://www.geocities.com/MarcJoye/publications.html`. Citations in this document: §1, §5.
28. Marc Joye, Sung-Ming Yen, *The Montgomery powering ladder*, in [29] (2003), 291–302. URL: `http://www.gemplus.com/smart/rd/publications/pdf/JY03mont.pdf`. Citations in this document: §8.
29. Burton S. Kaliski Jr., Çetin Kaya Koç, Christof Paar (editors), *Cryptographic hardware and embedded systems — CHES 2002, 4th international workshop, Redwood Shores, CA, USA, August 13–15, 2002, revised papers*, Lecture Notes in Computer Science, 2523, Springer-Verlag, 2003. ISBN 3–540–00409–2. See [28].
30. Çetin Kaya Koç, David Naccache, Christof Paar (editors), *Cryptographic hardware and embedded systems — CHES 2001, third international workshop, Paris, France, May 14–16, 2001, proceedings*, Lecture Notes in Computer Science, 2162, Springer, 2001. ISBN 3–540–42521–7. MR 2003g:94002. See [27], [33], [40].
31. Çetin Kaya Koç, Christof Paar (editors), *Cryptographic hardware and embedded systems, first international workshop, CHES'99, Worcester, MA, USA, August 12-13, 1999, proceedings*, Lecture Notes in Computer Science, 1717, Springer, 1999. ISBN 3–540–66646–X. See [18].
32. Tanja Lange, *Mathematical countermeasures against side-channel attacks*, in [16] (2005), 687–714. MR 2163785. Citations in this document: §8, §8.
33. Pierre-Yvan Liardet, Nigel P. Smart, *Preventing SPA/DPA in ECC systems using the Jacobi form*, in [30] (2001), 391–401. MR 2003k:94033. Citations in this document: §1, §5, §8.
34. Cláudio L. Lucchesi, Arnaldo V. Moura (editors), *LATIN'98: theoretical informatics*, Lecture Notes in Computer Science, 1380, Springer-Verlag, 1998. ISBN 3–540–64275–7. MR 99d:68007. See [4].
35. Victor S. Miller, *Use of elliptic curves in cryptography*, in [42] (1986), 417–426. MR 88b:68040. Citations in this document: §1.
36. Peter L. Montgomery, *Speeding the Pollard and elliptic curve methods of factorization*, Mathematics of Computation **48** (1987), 243–264. ISSN 0025–5718. MR 88e:11130. URL: `http://links.jstor.org/sici?sici=0025-5718(198701)48:177<243:STPAEC>2.0.CO;2-3`. Citations in this document: §1, §6, §7.
37. David Naccache, Pascal Paillier (editors), *Public key cryptography, 5th international workshop on practice and theory in public key cryptosystems, PKC 2002, Paris, France, February 12–14, 2002, proceedings*, Lecture Notes in Computer Science, 2274, Springer, 2002. ISBN 3–540–43168–3. MR 2005b:94044. See [12].
38. Nadia Nedjah, Luiza de Macedo Mourelle (editors), *Embedded Cryptographic Hardware: Methodologies & Architectures*, Nova Science Publishers, 2004. ISBN 1–59454–012–8. See [11].
39. Kazuo Ohta, Dingyi Pei (editors), *Advances in cryptology — ASIACRYPT'98: proceedings of the International Conference on the Theory and Application of Cryptology and Information Security held in Beijing*, Lecture Notes in Computer Science, 1514, Springer-Verlag, Berlin, 1998. ISBN 3–540–65109–8. MR 2000h:94002. See [17].
40. Elisabeth Oswald, Manfred Aigner, *Randomized addition-subtraction chains as a countermeasure against power attacks*, in [30] (2001), 39–50. MR 2003m:94068. Citations in this document: §8.

41. Bart Preneel, Stafford E. Tavares (editors), *Selected Areas in Cryptography, 12th International Workshop, SAC 2005, Kingston, ON, Canada, August 11–12, 2005, Revised Selected Papers*, Lecture Notes in Computer Science, 3897, Springer, 2006. ISBN 3–540–33108–5. MR 2007b:94002. See [1].
42. Hugh C. Williams (editor), *Advances in cryptology: CRYPTO '85*, Lecture Notes in Computer Science, 218, Springer, Berlin, 1986. ISBN 3–540–16463–4. MR 87d:94002. See [35].
43. Moti Yung, Yevgeniy Dodis, Aggelos Kiayias, Tal Malkin (editors), *9th international conference on theory and practice in public-key cryptography, New York, NY, USA, April 24–26, 2006, proceedings*, Lecture Notes in Computer Science, 3958, Springer, Berlin, 2006. ISBN 978–3–540–33851–2. See [7], [22].