Fast Isogeny Evaluation on Binary Curves

Gustavo Banegas, Nicolas Sarkis, and Benjamin Smith

LIX, CNRS, Inria, École Polytechnique, Institut Polytechnique de Paris
gustavo@cryptme.in, nicolas.sarkis@inria.fr, smith@lix.polytechnique.fr

Abstract. We give efficient formulas to evaluate isogenies of ordinary elliptic curves over finite fields of characteristic 2, extending the odd-characteristic techniques of Hisil–Costello and Renes to binary fields. For odd prime degree $\ell = 2s + 1$, our affine product evaluation computes the image x-coordinate using $5s\mathbf{M}$ field multiplications, or $4s\mathbf{M}$ when the kernel points are normalized. We derive an inversion-free variant that evaluates the x-map in projective and twisted Kummer coordinates, allowing carried points to remain projective across successive isogeny steps. Over $\mathbb{F}_{2^{511}}$, microbenchmarks show that the inversion-free projective and twisted variants are faster than Vélu-style x-evaluation when outputs are kept in projective/twisted form, while the affine one-inversion variant is about 4.2× faster.
Keywords: Isogeny-based cryptography – Binary elliptic curves – Kummer lines

1 Introduction

Elliptic curves have become ubiquitous in cryptography since their introduction forty years ago [23, 17]. They underpin a wide range of pre-quantum constructions, from encryption schemes to digital signatures and beyond. However, the advent of large-scale quantum computers threatens the security of these systems: Shor's algorithm solves the underlying discrete logarithm problem in polynomial time [32], motivating the search for *post-quantum* alternatives that remain secure against quantum adversaries.

Isogeny-based cryptography recycles elliptic curves to provide a basis for post-quantum cryptosystems. Important examples include the NIST signature standardization candidate SQISign [1] and the key exchange CSIDH [8]. These schemes are based on the evaluation of *isogenies*—morphisms of elliptic curves—and their efficiency critically depends on the efficiency of general elliptic curve arithmetic. This subject is thoroughly developed in odd characteristic, with standard models such as Montgomery curves [24] (or equivalently twisted Edwards models [5]) and efficient isogeny formulas [9, 27].

The characteristic-2 (binary) setting has an extensive literature on efficient computations on single curves: fast arithmetic formulas are available [35, 18, 26],

* Author list in alphabetical order; see https://ams.org/profession/leaders/CultureStatement04.pdf. This work was supported by the HYPERFORM consortium, funded by France through Bpifrance, and by the France 2030 program under grant agreement ANR-22-PETQ-0008 PQ-TLS. Date of this document: 2026-05-21.

and point counting is significantly faster in characteristic 2 [20]. However, general isogeny computations over binary fields have not yet been optimized. The lack of fast binary isogeny formulæ has proven a bottleneck in early investigations of binary isogeny-based cryptography [2]. Our work aims to fill this gap.

1.1 Explicit isogeny computations

Let $\mathcal{E}/\mathbb{F}_q$ be an elliptic curve. Given any finite $\mathbb{F}_q$-rational subgroup $G \subset \mathcal{E}$, there exists an elliptic curve $\mathcal{E}/G$ over $\mathbb{F}_q$ and a separable isogeny $\varphi : \mathcal{E} \to \mathcal{E}/G$ over $\mathbb{F}_q$ with kernel G. Both $\mathcal{E}/G$ and φ are only defined up to $\mathbb{F}_q$-isomorphism.

"Isogeny computation" generally refers to one of the following three problems:

Problem 1 (Isogeny codomain curve). Given $\mathcal{E}/\mathbb{F}_q$ and an $\mathbb{F}_q$-rational finite subgroup $G \subset \mathcal{E}$, compute a curve $\mathcal{E}_G/\mathbb{F}_q$ such that $\mathcal{E}_G \simeq \mathcal{E}/G$ over $\mathbb{F}_q$.

Problem 2 (Isogeny evaluation). Given $\mathcal{E}/\mathbb{F}_q$, an $\mathbb{F}_q$-rational finite subgroup $G \subset \mathcal{E}$, and points $P_1, \ldots, P_n$ in $\mathcal{E}(\mathbb{F}_q)$, compute the coordinates of $\varphi(P_1), \ldots, \varphi(P_n)$ where $\varphi : \mathcal{E} \to \mathcal{E}/G$ is a quotient isogeny.

In isogeny-based cryptosystems, the points $P_1, \ldots, P_n$ in Problems 2 and 3 are often bases of torsion subgroups that will be used to derive generators of kernels of further isogenies away from the codomain curve.

For many applications (especially in isogeny-based cryptography), it is enough to work on the associated Kummer line $\mathcal{E}/\pm 1$, or more concretely, with x-coordinates only. The Kummer line does not have a group structure, but it is still possible to compute scalar multiplication efficiently with the Montgomery ladder [24], using doubling and differential addition formulas—that is, formulas that compute $x(P+Q)$ given $x(P)$, $x(Q)$ and $x(P-Q)$. For isogeny formulas, we look for the x-coordinate of the image given the x-coordinate of the base point.

Problem 3 (x-only isogeny evaluation). Given $\mathcal{E}/\mathbb{F}_q$, an $\mathbb{F}_q$-rational finite subgroup $G \subset \mathcal{E}$, and values $x_1 = x(P_1), \ldots, x_n = x(P_n)$ in $\mathbb{F}_q$ for $P_1, \ldots, P_n$ on $\mathcal{E}$, compute $x(\varphi(P_1)), \ldots, x(\varphi(P_n))$ where $\varphi : \mathcal{E} \to \mathcal{E}/G$ is a quotient isogeny.

Remark 1. Problems 2 and 3 do *not* require knowledge or computation of an equation for the quotient curve $\mathcal{E}/G$ (that is, a solution to Problem 1). In fact, if we can solve Problem 2 for a few points P_i, then we can easily solve Problem 1 by interpolating the curve equation through the $\varphi(P_i)$. Likewise, for Weierstrass models, we can solve Problem 1 by solving Problem 3 on the x-coordinates of the 3-torsion or the 2-torsion (in odd characteristic).

Any isogeny φ defined over $\mathbb{F}_q$ can be easily factored into the composition of

1. a scalar multiplication;
2. a purely inseparable isogeny (i.e., some Frobenius isogeny); and
3. a series of separable isogenies of prime degree, all defined over $\mathbb{F}_q$.

For scalar multiplication Problem 1 is vacuous, and there is a vast literature on efficient scalar multiplication (i.e., solving Problems 2 and 3). All three problems are near-trivial for Frobenius isogenies (Galois conjugation gives the codomain curve and point images).

In the case of separable prime-degree isogenies, however, Problems 1, 2, and 3 become interesting. All three can all be solved using Vélu's formulae [36], which assume a rational generator for G. A more general version of Vélu appears in [19, §2.4], where G is presented as the *kernel polynomial* $f_G(x)$, i.e. the minimal polynomial such that $f_G(x(Q)) = 0$ for all $Q \neq 0$ in G. Both versions of Vélu's approach require $O(\ell)$ $\mathbb{F}_q$-operations, where ℓ is the prime degree of the isogeny.

The formulæ of [36] are *additive*, in the sense that they express the symmetric functions defining the rational maps of the isogeny as sums, with each term corresponding to an element of G. The formulæ in [19, §2.4], attributed to Vélu, are derived from the same additive definition but are rewritten in terms of the kernel polynomial $\psi_G(x)$ (and its derivatives), yielding explicit expressions for x_G and y_G. Equivalently, instead of describing Vélu's equations in terms of the coordinates of the points of G, Kohel formulates them in terms of a generator of the ideal sheaf of G; thus, constructing the isogeny reduces to producing a single generator polynomial for that ideal sheaf (namely ψ_G).

Hisil and Costello [9] observed that these formulæ can be evaluated more efficiently when written in *multiplicative* form: that is, as a product of linear terms, each corresponding to an element of G. These expressions are particularly simple for odd ℓ on Montgomery models of elliptic curves (in odd characteristic), where they can exploit the near-diagonal action on coordinates of translation by the special 2-torsion point (Renes [27] gives a detailed exposition of the theory and some generalizations). Using this idea, they showed significant practical speedups for Problem 3—while maintaining the same $O(\ell)$ asymptotic complexity—in the case where G has an $\mathbb{F}_q$-*rational* generator. This $O(\ell)$ can be reduced to $\widetilde{O}(\sqrt{\ell})$ $\mathbb{F}_q$-operations (at least when q is odd) using the *Vélusqrt* algorithm [6]. In a different direction, if the generator of G is only defined over an extension of $\mathbb{F}_q$ then we can exploit Galois structures to reduce the cost of the Hisil–Costello formulae using the method of [3].

In the Couveignes–Rostovtsev–Stolbunov (CRS) framework and its refinements [11, 30, 12], isogeny evaluation is performed repeatedly to *push forward* points along chains, so the overall performance is often dominated by the cost of these evaluations. Over binary fields $\mathbb{F}_{2^m}$, however, the available odd-degree isogeny formulas are essentially specialized Vélu formulas, which are not designed to be optimal for x-only arithmetic or for inversion-free projective workflows. This drives the development of $\mathbb{F}_{2^m}$-specific isogeny evaluation formulas that reduce multiplications and, crucially, better support projective (and twisted) coordinates for point pushforward.

1.2 Contributions

We work with ordinary binary elliptic curves in the form

$$\mathcal{E}: \; y^2 + xy = x(x^2 + ax + b^2),$$

with $a, b \in \mathbb{F}_{2^m}$ and $b \neq 0$. Any ordinary curve over $\mathbb{F}_{2^m}$ is isomorphic to such a curve (as we will see in §4, isogenies in the binary supersingular case are easy to deal with). A key feature of this model is that the unique non-trivial 2-torsion point is normalized to $T = (0, 0)$, and translation by this point is particularly simple; the resulting symmetry gives faster curve arithmetic. Further, the points of order 4 above T have rational x-coordinate b, which allows some simplifications when working on the Kummer line in twisted coordinates $(\widetilde{X} : \widetilde{Z}) := (bZ : X)$.

First, we extend the product-style viewpoint of Renes to characteristic 2 (Theorem 2), obtaining a multiplicative formula for the x-coordinate map of odd-degree isogenies that is naturally compatible with projective computation. For odd prime degrees ℓ, we instantiate the required auxiliary point outside the kernel using the non-trivial 2-torsion point T, which is always available on ordinary curves in this model.

Second, specializing to $\mathcal{E}$ and exploiting the explicit translation-by-T structure, we derive concrete projective evaluation formulas for the x-map (Propositions 3 and 4). These formulas avoid inversions during the kernel-product loop and, crucially, allow carrying Kummer coordinates $(X : Z)$ (or twisted $(\widetilde{X} : \widetilde{Z})$) across successive steps. We provide an explicit operation-count analysis for x-evaluation as a function of ℓ, with and without kernel-point normalization. In particular, while we compute codomains using Vélu-style formulas (to benefit from normalization of kernel points across steps), the evaluation itself is performed inversion-free in projective/twisted form. The resulting costs are summarized in Table 1.

Finally, to complement the operation-count model, we report microbenchmarks of our evaluation formulas. We implement the competing methods and time the computation of $x(\varphi(P))$ for randomly sampled inputs P, averaging over many repeated calls per prime degree ℓ (and over multiple random kernels) to reduce measurement noise; full details are given in Section 8[1].

2 Arithmetic in binary fields

Throughout this paper, we work over the binary field $\mathbb{F}_{2^m}$. We will need a few elementary facts on arithmetic in $\mathbb{F}_{2^m}$. The first is that every element has a unique square root, given by

$$\sqrt{x} = x^{2^{m-1}}.$$

[1] Our implementation is available for download from https://gitlab.inria.fr/gsouzaba/iso_binary.

Table 1: Cost of evaluating an ℓ-isogeny on the level of Kummer lines (for odd ℓ), in $\mathbb{F}_{2^m}$-operations, with $s = (\ell - 1)/2$. Normalizing kernel points (scaling one coordinate to 1) adds a precomputation of $3(s-3)\mathbf{M} + 1\mathbf{I}$ (see §7.1), but this cost can be amortized over many evaluations of the same isogeny. The cost of generating the kernel points (which, similarly, can be amortized over several isogeny evaluations) is not counted here. $\mathbf{M}$, $\mathbf{S}$, $\mathbf{m}_0$, and $\mathbf{I}$ are defined in §2.3.

	Kernel points		
Coordinates	not normalized	normalized	Proof
Projective	$5s\mathbf{M} + 2\mathbf{S} + (2+s)\mathbf{m}_0$	$4s\mathbf{M} + 2\mathbf{S} + 3\mathbf{m}_0$	Prop. 3
Twisted projective	$5s\mathbf{M} + 2\mathbf{S}$	$4s\mathbf{M} + 2\mathbf{S}$	Prop. 4

2.1 The trace map

The Frobenius automorphism $\pi : x \mapsto x^2$ generates the Galois group of $\mathbb{F}_{2^m}/\mathbb{F}_2$. The *trace* is the linear map $\mathrm{Tr} : \mathbb{F}_{2^m} \to \mathbb{F}_2$ defined by

$$\mathrm{Tr}(x) := \sum_{i=0}^{m-1} \pi^i(x) = \sum_{i=0}^{m-1} x^{2^i} \qquad \text{for } x \in \mathbb{F}_{2^m}\,.$$

By definition, $\mathrm{Tr} \circ \pi = \pi \circ \mathrm{Tr}$, so $\mathrm{Tr}(x)^2 = \mathrm{Tr}(x^2) = \mathrm{Tr}(x)$.

Lemma 1. *There exists an element ε in $\mathbb{F}_{2^m}$ such that $\mathrm{Tr}(\varepsilon) = 1$. In particular, if z is a generator of $\mathbb{F}_{2^m}$ (so $\mathbb{F}_{2^m} = \mathbb{F}_2(z)$), then we can take $\varepsilon = z^i$ for some $0 \le i < m$.*

Proof. The trace is a homomorphism (of $\mathbb{F}_2$-vector spaces) into $\mathbb{F}_2 = \{0, 1\}$, but is not identically zero (because $\mathbb{F}_{2^m}/\mathbb{F}_2$ is finite and separable; see e.g. [34, Lemma 0BIL]), so it is surjective. If $\mathrm{Tr}(z^i) = 0$ for every $0 \le i < m$, then by $\mathbb{F}_2$-linearity the trace would be zero everywhere, a contradiction; hence $\mathrm{Tr}(z^i) = 1$ for some $0 \le i < m$. □

Definition 1. *For any binary field $\mathbb{F}_{2^m}$, we fix an element ε such that $\mathrm{Tr}(\varepsilon) = 1$ as in Lemma 1. (If m is odd, then we can take $\varepsilon = 1$.)*

2.2 Solving quadratic equations

We will need to solve quadratic equations $ax^2 + bx + c = 0$ over $\mathbb{F}_{2^m}$, with $a \neq 0$ [15, § 1.4]. There are two cases:

1. If $b = 0$ then there is one solution, namely $\sqrt{\frac{c}{a}}$.
2. Otherwise, setting $y = \frac{ax}{b}$, the equation is equivalent to $y^2 + y = \delta$ with $\delta = \frac{ca}{b^2}$. Since $\mathrm{Tr}(y) = \mathrm{Tr}(y^2)$:
   - If $\mathrm{Tr}(\delta) = 1$, then the equation has no solution.

– Otherwise, $\mathrm{Tr}(\delta) = 0$. Fixing $\varepsilon \in \mathbb{F}_{2^m}$ such that $\mathrm{Tr}(\varepsilon) = 1$, the two solutions to the quadratic equation are $H(\delta)$ and $H(\delta) + 1$, where

$$H(\delta) := \sum_{i=0}^{m-1} \sum_{j=0}^{i} \varepsilon^{2^j} \delta^{2^i} .$$

In particular, if m is odd, then we can take $\varepsilon = 1$ and $H(\delta)$ is the *half-trace*

$$H(\delta) = \sum_{i=0}^{(m-1)/2} \delta^{2^{2i}}.$$

Following Pornin's notation [26], we let $\mathrm{QSolve}(\delta)$ denote one solution to the equation $x^2 + x = \delta + \varepsilon \,\mathrm{Tr}(\delta)$, i.e. $\mathrm{QSolve}(\delta) = H(\delta + \varepsilon \,\mathrm{Tr}(\delta))$.

These operations have negligible cost compared to multiplication, since they are linear. Once an $\mathbb{F}_2$-basis of $\mathbb{F}_{2^m}$ is fixed, it is enough to pre-compute the corresponding $\mathbb{F}_2$-matrix and then apply the linear operation. For example: if $x = \sum_{i=0}^{510} x_i z^i$ in $\mathbb{F}_{2^{511}} = \mathbb{F}_2[z]/(z^{511} + z^{10} + 1)$, then $\mathrm{Tr}(x) = x_0 + x_{501}$.

2.3 Field operations

We use the following notation for elementary operations in $\mathbb{F}_{2^m}$:

– $\mathbf{M}$ denotes the cost of a generic field multiplication.
– $\mathbf{S}$ denotes the cost of a field squaring; in characteristic 2, squarings are typically much cheaper than multiplications (for example, see [21, Tab. 2], where squarings are 5 to 7 times faster than multiplications).
– $\mathbf{m}_0$ is the cost of multiplication by a curve constant; we count $\mathbf{m}_0$ separately from $\mathbf{M}$ since curve parameters can sometimes be chosen so that these constants admit faster multiplication. However, in isogeny-based cryptographic applications the curve parameter is usually out of our control, so $\mathbf{m}_0 = \mathbf{M}$.
– $\mathbf{I}$ denotes the cost of a field inversion.

3 Binary elliptic curves

In this section, we recall some facts on elliptic curves, their Kummer lines, and isogenies over binary fields $\mathbb{F}_{2^m}$.

3.1 Weierstrass equations and the Kummer line

Every elliptic curve $\mathcal{E}/\mathbb{F}_{2^m}$ can be defined by a Weierstrass equation

$$\mathcal{E} : y^2 + a_1 xy + a_3 y = x^3 + a_2 x^2 + a_4 x + a_6 \tag{1}$$

with a_1, a_2, a_3, a_4, and a_6 in $\mathbb{F}_{2^m}$. We note that a_1 and a_3 cannot both be 0 (otherwise the curve is singular). We also work with the projective model

$$\mathcal{E} : Y^2 Z + a_1 XYZ + a_3 YZ^2 = X^3 + a_2 X^2 Z + a_4 XZ^2 + a_6 Z^3 ,$$

which is related to the affine model by $x = X/Z$, $y = Y/Z$.

The curve $\mathcal{E}$ has a natural geometric group law where the point at infinity $0_\mathcal{E} = (0 : 1 : 0)$ is the neutral element. (We will not need the general addition formulae here, but they can be found in [33, Alg. III.2.3]). The negation map

$$[-1] : (x, y) \longmapsto (x, y + a_1 x + a_3)$$

(with $0_\mathcal{E} \mapsto 0_\mathcal{E}$) fixes precisely the points in the 2-torsion subgroup $\mathcal{E}[2]$.

The quotient of $\mathcal{E}$ by the action of $[-1]$ is the *Kummer line* $\mathcal{E}/\langle\pm 1\rangle$, which is isomorphic to the projective line $\mathbb{P}^1$. The projection $\mathcal{E} \to \mathcal{E}/\langle\pm 1\rangle \simeq \mathbb{P}^1$ is defined by $P = (X_P : Y_P : Z_P) \mapsto x(P) = (X_P : Z_P)$ when $Z_P \neq 0$, and $0_\mathcal{E} = (0 : 1 : 0) \mapsto (1 : 0)$.

The Kummer line does not have a group law, but the scalar multiplication $x(P) \mapsto x([n]P)$ is well-defined, and can be computed with the Montgomery ladder (see [24] and [10], and in particular [25] for the ladder on binary curves). This relies on two operations:

- Differential addition, denoted `xADD`: given $x(P)$, $x(Q)$ and $x(P-Q)$, compute $x(P+Q)$;
- Doubling, denoted `xDBL`: given $x(P)$, compute $x([2]P)$.

We will study several models of binary elliptic curves and their associated Kummer lines.

Remark 2. In [28, 29] Robert and Sarkis studied these operations on Kummer lines extensively in odd characteristic to derive isogeny formulas, and Kummer lines more generally in [4] along with Barbulescu. Unfortunately, their construction does not extend to characteristic 2 because it relies on having four 2-torsion points, which is never the case for binary elliptic curves.

3.2 Isomorphisms

Proposition 1. *If $\mathcal{E}$ and $\mathcal{E}'$ are Weierstrass models over a binary field $\mathbb{F}_{2^m}$ with coordinates (x, y), (x', y') and constants a_i, a'_i respectively, then every isomorphism $\mathcal{E} \to \mathcal{E}'$ corresponds to a tuple (u, r, s, t) in $\mathbb{F}_{2^m}^4$ with $u \neq 0$ such that*

$$x = u^2 x' + r \qquad \text{and} \qquad y = u^3 y' + u^2 s x' + t$$

and

- $u a'_1 = a_1$;
- $u^2 a'_2 = a_2 + s a_1 + r + s^2$;
- $u^3 a'_3 = a_3 + r a_1$;
- $u^4 a'_4 = a_4 + s a_3 + (t + rs) a_1 + r^2$;
- $u^6 a'_6 = a_6 + r a_4 + r^2 a_2 + r^3 + t a_3 + t^2 + rt a_1$.

Proof. See [33, Prop. III.3.1b, Tab. III.3.1].

We will often refer to an isomorphism via the tuple (u, r, s, t). In particular, when $a_1 \neq 0$, we can always reduce to the form

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

with $a, b \in \mathbb{F}_{2^m}$ and $b \neq 0$. The j-invariant of this model is $1/b$, so a only controls the quadratic twist.

3.3 2-torsion of binary elliptic curves

As mentioned above, the negation map

$$[-1] : (x, y) \longmapsto (x, y + a_1 x + a_3)$$

(with $0_{\mathcal{E}} \mapsto 0_{\mathcal{E}}$) fixes precisely the points in the 2-torsion subgroup $\mathcal{E}[2]$. There are two possibilities:

- If $a_1 = 0$, then the only fixed point is $0_{\mathcal{E}}$, so $\mathcal{E}[2] = \{0_{\mathcal{E}}\} \simeq 0$; the curve $\mathcal{E}$ is **supersingular**.
- If $a_1 \neq 0$, then $[-1]$ has two fixed points: $0_{\mathcal{E}}$ and

$$T := (x_T, y_T) \quad \text{where} \quad \begin{cases} x_T & := a_3/a_1\,, \\ y_T & := \sqrt{x_T^3 + a_2 x_T^2 + a_4 x_T + a_6} \end{cases}$$

  (the square root exists in $\mathbb{F}_{2^m}$, and is unique, because $\mathbb{F}_{2^m}$ has characteristic 2). We say $\mathcal{E}$ is **ordinary**.

Suppose $\mathcal{E}$ is ordinary, and let T be the unique 2-torsion point defined above. The translation-by-T map on $\mathcal{E}$ is defined on x-coordinates by

$$x_P \longmapsto x_{P+T} = \frac{x_T x_P + a_1 y_T + a_4}{x_P + x_T}.$$

for all $P \neq 0_{\mathcal{E}}, T$. There are exactly two 4-torsion points above T, namely $\widetilde{T}$ and $-\widetilde{T} = \widetilde{T} + T$. Their shared x-coordinate is

$$x_{\widetilde{T}} = \sqrt{a_1 y_T + a_4}\,.$$

(Note that $x_{\widetilde{T}}$ is always $\mathbb{F}_{2^m}$-rational.)

Example 1. The case $a_3 = 0$ is very useful for us. There, $T = (0, \sqrt{a_6})$ and

$$x_{P+T} = \frac{a_4 + a_1\sqrt{a_6}}{x_P}\,, \tag{2}$$

so the x-coordinate of the 4-torsion points satisfies $x_{\widetilde{T}} = (a_4 + a_1\sqrt{a_6})/x_{\widetilde{T}}$, so

$$x_{\widetilde{T}} = \sqrt{a_4 + a_1\sqrt{a_6}}\,. \tag{3}$$

3.4 Curve models with 2-torsion

In this section, we will focus on models for ordinary curves. The most basic is

$$y^2 + xy = x^3 + a_2x^2 + a_6 \tag{4}$$

where $a_2, a_6 \in \mathbb{F}_{2^m}$, and $a_6 \neq 0$. To ease computations and have more efficient formulas, we can consider isomorphisms from Proposition 1 to change the model of the curve. Ultimately, we want to derive model-preserving isogeny formulæ, since (especially for cryptographic applications) we want to compute chains of isogenies between the same kind of curves.

The map $(x, y) \longmapsto (x, y + \sqrt{a_6})$ takes us from the curve of Eq. (4) to the following model studied by Kohel [18] and Pornin [26]:

$$\mathcal{E} : y^2 + xy = x(x^2 + ax + b^2) \tag{5}$$

with $a = a_2$ and $b = a_6^{1/4} \in \mathbb{F}_{2^m}$, moving the 2-torsion point to $(0, 0)$. It is important to note that this isomorphism doesn't change the x-coordinate, so applying it is cost-free at the Kummer line level.

The j-invariant of this model is $1/b^4$: once again a only controls the quadratic twist, so we can reduce to the two curves

$$\mathcal{E}_0 : y^2 + xy = x(x^2 + b^2) \quad \text{and} \quad \mathcal{E}_1 : y^2 + xy = x(x^2 + \varepsilon x + b^2) \tag{6}$$

where $\varepsilon \in \mathbb{F}_{2^m}$ is an element of trace 1 (as in Definition 1). Starting from Eq. (5), the isomorphism given by $(u, r, s, t) = (1, 0, \mathrm{QSolve}(a), 0)$ goes to $\mathcal{E}_{\mathrm{Tr}(a)}$.

The 2-torsion on $\mathcal{E}_0$ and $\mathcal{E}_1$ is generated by $T = (0, 0)$ (this will be convenient in Sections 5 and 6), and by Eq. (3) the two points of order 4 above T have x-coordinate b.

- On $\mathcal{E}_0$, the points of order 4 are $\widetilde{T} = (b, 0)$ and $-\widetilde{T} = (b, b)$: in particular, they are rational, so $\mathcal{E}_0[4] = \mathcal{E}_0[4](\mathbb{F}_{2^m}) = \{0_\mathcal{E}, T, (b, 0), (b, b)\}$.
- On $\mathcal{E}_1$, the points of order 4 are $\widetilde{T} = (b, b\omega)$ and $-\widetilde{T} = (b, b\omega + b)$ where $\omega \in \mathbb{F}_{2^{2m}}$ satisfies $\omega^2 + \omega = \varepsilon$: in particular, these points are irrational, so $\mathcal{E}_1[4](\mathbb{F}_{2^m}) = \{0_\mathcal{E}, T\}$ and $\mathcal{E}_1[4] = \mathcal{E}_1[4](\mathbb{F}_{2^{2m}}) = \{0_\mathcal{E}, T, (b, b\omega), (b, b(\omega + 1))\}$.

The twisting isomorphism $\mathcal{E}_0 \to \mathcal{E}_1$ is $(x, y) \mapsto (x, y + b\omega)$ over $\mathbb{F}_{2^{2m}}$.

Remark 3. Kohel [18] actually looks for normal forms in characteristic 2 similar to Edwards normal forms by studying their symmetries. His $\mathbb{Z}/4\mathbb{Z}$- and split μ_4-normal forms are in the form $\mathcal{E}_0$ for some $b \in \mathbb{F}_{2^m}$. These are a special case of binary Edwards curves in the sense of [7], which provide complete addition formulas for any ordinary elliptic curve.

3.5 Arithmetic on Kummer lines

Kummer-line arithmetic relies on the two operations

$$\texttt{xADD} : \big((X_P : Z_P), (X_Q : Z_Q), (X_{P-Q} : Z_{P-Q})\big) \longmapsto (X_{P+Q} : Z_{P+Q}) \tag{7}$$

$$\texttt{xDBL} : (X_P : Z_P) \longmapsto (X_{[2]P} : Z_{[2]P}) \tag{8}$$

We will now provide explicit formulæ for Kummer arithmetic on each of our curve models so that the article is self-contained. When counting operations, we implicitly use the identity

$$AB + CD = (A + C)(B + D) + AD + BC \tag{9}$$

to compute sums of products using $3\mathbf{M}$ instead of $4\mathbf{M}$.

Kohel–Pornin models. The Kummer operations on the curve $\mathcal{E}$ of Eq. (5) is defined by

$$\texttt{xADD} : \begin{cases} X_{P+Q} = X_{P-Q}(X_P Z_Q + X_Q Z_P)^2 + Z_{P-Q}(X_P Z_Q)(X_Q Z_P) \\ Z_{P+Q} = Z_{P-Q}(X_P Z_Q + X_Q Z_P)^2 \end{cases} \tag{10}$$

which can be performed in $6\mathbf{M} + 1\mathbf{S}$, and

$$\texttt{xDBL} : \begin{cases} X_{[2]P} = (X_P + bZ_P)^4 \\ Z_{[2]P} = (X_P Z_P)^2 \end{cases} \tag{11}$$

which can be performed in $1\mathbf{M} + 3\mathbf{S} + 1\mathbf{m}_0$. A full step of the Montgomery ladder then costs $7\mathbf{M} + 4\mathbf{S} + 1\mathbf{m}_0$. These are the formulæ used by Pornin [26, §5.4]. They were originally derived by López and Dahab [21, Eqs. (8)-(9)] for the model of Eq. (4) with $a_2 = a$ and $a_6 = b^4$; the isomorphisms from this curve to $\mathcal{E}_0$ and $\mathcal{E}_1$ do not change the x-coordinate, so the `xADD` and `xDBL` formulæ for all these curves are the same.

Remark 4. In contexts where we stay on the same curve and the same base point is re-used as $P - Q$ in `xADD`, it makes sense to normalize Z_{P-Q} to 1, so multiplication by Z_{P-Q} is free. In some cases, we can even choose X_{P-Q} to be small, so multiplications by X_{P-Q} cost less than $\mathbf{M}$. But we cannot make these assumptions when chaining isogenies across different curves, so in this paper we count multiplications by X_{P-Q} as generic $\mathbf{M}$ (and the same for Z_{P-Q}, when it is not normalized to 1).

Stam's model. Stam [35, § 3.2], provides alternative formulas with fewer multiplications by projective coordinates of $P - Q$ on the curve

$$\mathcal{E} : y^2 + \frac{1}{b}xy = x^3 + ax^2 + b^2 \tag{12}$$

with $b \neq 0$. The Kummer operations for this model are defined by

$$\texttt{xADD} : \begin{cases} X_{P+Q} = Z_{P-Q}(X_P X_Q + Z_P Z_Q)^2 \,, \\ Z_{P+Q} = X_{P-Q}(X_P Z_Q + X_Q Z_P)^2 \,, \end{cases} \tag{13}$$

which can be computed in $5\mathbf{M} + 2\mathbf{S}$, and

$$\texttt{xDBL} : \begin{cases} X_{[2]P} = \left(b(X_P^2 + Z_P^2)\right)^2 \,, \\ Z_{[2]P} = (X_P Z_P)^2 \,, \end{cases} \tag{14}$$

in $1\mathbf{M} + 3\mathbf{S} + 1\mathbf{m}_0$, for a total ladder-step cost of $6\mathbf{M} + 5\mathbf{S} + 1\mathbf{m}_0$, beating the López–Dahab formulas. Kohel's split μ_4-normal form achieves the same operation count [18, Cor. 26].

Twisted Stam coordinates. If we want to go to a curve shaped as in Eq. (5), we can consider the isomorphism given by $(u, r, s, t) = (1/b, 0, 0, b)$, mapping $\mathcal{E}$ to

$$\mathcal{E}' : y'^2 + x'y' = x' \left(x'^2 + ab^2x' + b^4\right) = x' \left(x'^2 + a\beta x' + \beta^2\right)$$

with $\beta = b^2$. The x-coordinate is modified by the isomorphism $x \mapsto x' := b^2x = \beta x$, which suggests using *twisted* coordinates as we will see below. First, consider the `xADD` and `xDBL` formulas on $\mathcal{E}'$, losing $3\mathbf{m}_0$ on `xADD`, where $x' = X/Z$:

$$\texttt{xADD} : \begin{cases} X_{P+Q} = \beta^2 Z_{P-Q}(X_P X_Q + (\beta Z_P)(\beta Z_Q))^2 \\ Z_{P+Q} = X_{P-Q}(X_P(\beta Z_Q) + X_Q(\beta Z_P))^2 \end{cases} \tag{15}$$

and

$$\texttt{xDBL} : \begin{cases} X_{[2]P} = (X_P^2 + \beta^2 Z_P^2)^2 \\ Z_{[2]P} = (X_P Z_P)^2 \end{cases} \tag{16}$$

Gaudry–Lubicz models. Gaudry and Lubicz studied Kummer lines [14, § 6.1] in characteristic 2 using the theta group theory and algebraic theta functions. Based on their work, Karati then proposed a concrete binary Kummer line [16], associated to the curve

$$\mathcal{E} : y^2 + xy = x^3 + b^4.$$

It is isomorphic to $\mathcal{E}_0$ via $(u, r, s, t) = (1, 0, 0, b^2)$, hence it has full rational 4-torsion. Given that the x-coordinate is not changed, if we have formulas on $\mathcal{E}$ then we will have formulas for $\mathcal{E}_0$ too.

Gaudry and Lubicz gave formulas for their Kummer line which are the same as Stam's from Eqs. (13) and (14). This is no surprise, Karati [16, § 2.2] introduced a twisted coordinate $(\widetilde{X} : \widetilde{Z}) := (bZ : X)$, or in an affine version $\widetilde{x} := b/x$. This corresponds – up to an inverse — to the x-coordinate on an associated Stam curve (Eq. (12)). Also, with these formulas, if T is the non-trivial 2-torsion point, then $x_{P+T} = b^2/x_P$ implies that $\widetilde{x}_{P+T} = 1/\widetilde{x}_P$.

The formulas for `xADD` and `xDBL` with twisted coordinates — derived from Eqs. (13) and (14) — are then as follows:

$$\texttt{xADD} : \begin{cases} \widetilde{X}_{P+Q} = \widetilde{Z}_{P-Q}(\widetilde{X}_P\widetilde{Z}_Q + \widetilde{X}_Q\widetilde{Z}_P)^2 \\ \widetilde{Z}_{P+Q} = \widetilde{X}_{P-Q}(\widetilde{X}_P\widetilde{X}_Q + \widetilde{Z}_P\widetilde{Z}_Q)^2 \end{cases} \tag{17}$$

and

$$\texttt{xDBL} : \begin{cases} \widetilde{X}_{[2]P} = (\widetilde{X}_P\widetilde{Z}_P)^2 \\ \widetilde{Z}_{[2]P} = b(\widetilde{X}_P^2 + \widetilde{Z}_P^2)^2 \end{cases} \tag{18}$$

The cost of a complete step is still $6\mathbf{M} + 5\mathbf{S} + 1\mathbf{m}_0$. We will see in the next section that using twisted coordinates is also convenient isogeny-wise as it saves some multiplications by the curve constant b, hence we will include those in our comparisons.

Remark 5. Karati has a slightly different approach, he first introduces the same formulas as in Eqs. (13) and (14) as the `xADD` and `xDBL` formulas [16, Tab. 1], just as Gaudry and Lubicz did. He then explains in Section 2.3 how to correct those to get a functional scalar multiplication by correcting by a 2-torsion point. In particular, the map $\widehat{\pi}^{-1}$ maps a point $P = (X : \cdot : Z) \in \mathcal{E}/\pm 1$ to $(X : bZ) = (\widetilde{Z} : \widetilde{X})$. Therefore, his Equation (9) can be read as

$$\begin{cases} (\widetilde{Z}_{2\cdot P} : \widetilde{X}_{2\cdot P}) & = \texttt{xDBL}((\widetilde{Z}_P : \widetilde{X}_P)), \\ (\widetilde{Z}_{P+Q} : \widetilde{X}_{P+Q}) & = \texttt{xADD}((\widetilde{Z}_P : \widetilde{X}_P), (\widetilde{Z}_Q : \widetilde{X}_Q), (\widetilde{Z}_{P-Q} : \widetilde{X}_{P-Q})), \end{cases}$$

where `xDBL` and `xADD` correspond to his formulas from Table 1. This yields the formulas from Eqs. (17) and (18).

The idea of using twisted coordinates was also used by Pornin [26, § 3, § 5.4], although putting formulas from Eqs. (10) and (11) in twisted coordinates adds a multiplication by b at each step of the ladder.

4 The supersingular case

The case of supersingular binary curves is particularly simple, because there is essentially only one such curve—so all isogenies are isomorphic to, or at worst twists of, endomorphisms. The endomorphism ring can easily be made explicit for this curve, so all isogeny computations can be reduced (via isomorphisms and twists) to sums of scalar multiples of automorphisms.

Consider the curve $\mathcal{E} : y^2 + y = x^3$. Write $\mathbb{F}_4 = \mathbb{F}_2(\omega)$, where $\omega^2 + \omega + 1 = 0$ (i.e., ω is a primitive 3rd root of unity). Over $\mathbb{F}_4$, the automorphism group of $\mathcal{E}$ is

$$\mathrm{Aut}(\mathcal{E}) = \langle \alpha, \beta, \gamma, [-1] \rangle ,$$

where

- $[-1] : (x, y) \mapsto (x, y + 1)$ is the usual negation map;
- $\alpha : (x, y) \mapsto (\omega x, y)$ has order 3;
- $\beta : (x, y) \mapsto (x + 1, y + x + \omega^2)$ has order 4;
- $\gamma : (x, y) \mapsto (x + \omega, y + \omega^2 x + \omega^2)$ has order 4;
- $\beta^2 = \gamma^2 = [-1]$, $\beta\gamma = [-1]\gamma\beta$, and $\beta\alpha = \alpha\gamma$.

In particular, α does not commute with the 2-power Frobenius endomorphism π_2 (we have $\pi_2\alpha = -\alpha\pi_2$), so $\mathrm{End}(\mathcal{E})$ is not commutative, and $\mathcal{E}$ is supersingular. Indeed, $\mathrm{End}(\mathcal{E})$ is a quaternion algebra (see [33, Thm. V.3.1a]), generated by the automorphisms above. In particular, any endomorphism of $\mathcal{E}$ can be written as a $\mathbb{Z}$-linear combination of these automorphisms.

Proposition 2. *An elliptic curve $\mathcal{E}/\mathbb{F}_{2^m}$ is supersingular if and only if $j(\mathcal{E}) = 0$, if and only if $\mathcal{E}$ is isomorphic over $\overline{\mathbb{F}}_{2^m}$ to the curve defined by $y^2 + y = x^3$.*

More specifically, if $t_\mathcal{E} = 2^m + 1 - \#\mathcal{E}(\mathbb{F}_{2^m})$ (i.e. $t_\mathcal{E}$ is the trace of Frobenius), then the possible values for $t_\mathcal{E}$, and the number of $\mathbb{F}_{2^m}$-isomorphism classes of curves with that trace, are the following:

– $t_{\mathcal{E}} = 0$*: 1 isomorphism class.*
– $t_{\mathcal{E}} = 2^{m/2}$ *(only if m is even): 2 isomorphism classes.*
– $t_{\mathcal{E}} = -2^{m/2}$ *(only if m is even): 2 isomorphism classes.*
– $t_{\mathcal{E}} = 2^{(m+1)/2}$ *(only if m is odd): 1 isomorphism class.*
– $t_{\mathcal{E}} = -2^{(m+1)/2}$ *(only if m is odd): 1 isomorphism class.*
– $t_{\mathcal{E}} = 2^{(m+2)/2}$ *(only if m is even): 1 isomorphism class.*
– $t_{\mathcal{E}} = -2^{(m+2)/2}$ *(only if m is even): 1 isomorphism class.*

Proof. See [22, § 3.4–3.6, Tab. 3.4].

The upshot is that if $\varphi : \mathcal{E}_1 \to \mathcal{E}_2$ is an isogeny of supersingular curves over $\mathbb{F}_{2^m}$, then $\varphi = \tau_2^{-1} \circ \psi \circ \tau_1$ where $\tau_1 : \mathcal{E}_1 \to \mathcal{E}$ and $\tau_2 : \mathcal{E}_2 \to \mathcal{E}$ are isomorphisms (possibly defined over an extension) and ψ is an endomorphism of the curve $\mathcal{E}$ above; so the isogeny φ can be efficiently computed as a $\mathbb{Z}$-linear combination of the generators of $\mathrm{End}(\mathcal{E})$.

5 2-isogenies of ordinary binary curves

Let $\varphi : \mathcal{E} \to \mathcal{E}'$ be an isogeny of binary curves. Writing $\deg(\varphi) = 2^r m$ for some $r \geq 0$ and odd m, we can factor φ into the composition of an m-isogeny and a series of 2-isogenies. For binary curves, 2-isogenies are particularly simple to handle, so we will deal with them first before developing algorithms for the odd-degree isogenies in §6.

Up to isomorphism, there are only one or two 2-isogenies from a binary curve $\mathcal{E}$. The first, and most fundamental, is the 2-power Frobenius $\pi : (x, y) \mapsto (x^2, y^2)$; if $\mathcal{E}$ is supersingular, then this is the only 2-isogeny.

If $\mathcal{E}$ is ordinary, then there is also the quotient isogeny with kernel $\langle T \rangle$. We start by presenting the normalized isogeny for the general ordinary Weierstrass model.

Example 2. If $\mathcal{E}$ is defined by a Weierstrass model $y^2 + xy = x^3 + a_2x^2 + a_4x + a_6$, then the distinguished 2-torsion point is $T = (0, \sqrt{a_6})$ and the quotient curve is

$$\mathcal{E}/\langle T \rangle \simeq \mathcal{E}' : v^2 + uv = u^3 + a_2u^2 + \sqrt{a_6}u + (a_6 + c)$$

where $c = a_4 + \sqrt{a_6}$, and the normalized isogeny $\varphi_2 : \mathcal{E} \to \mathcal{E}'$ is defined by

$$\varphi_2 : (x, y) \longmapsto (u, v) \quad \text{where} \quad \begin{cases} u = (x^2 + c)/x\,, \\ v = y(u/x) + \frac{c(x+\sqrt{a_6})}{x^2}\,. \end{cases}$$

The induced map of Kummer lines is $(X_P : Y_P) \mapsto (X_P^2 + cZ_P^2 : X_PZ_P)$.

For the optimized curve models, we compose the normalized 2-isogeny with a suitable isomorphism to get a 2-isogeny $\varphi : \mathcal{E} \to \mathcal{E}'$ where $\mathcal{E}'$ is presented in the same special model. In each of Examples 3, 4, and 5, the 2-power Frobenius isogeny π_2 maps $\mathcal{E}'$ back into $\mathcal{E}$, and in fact $\pi_2\varphi_2 = [2]_{\mathcal{E}}$. Indeed, in each case, composing the Kummer-line map with Frobenius gives the corresponding `xDBL` formula.

Example 3. On the Kohel and Pornin model $\mathcal{E} : y^2 + xy = x(x^2 + ax + b^2)$, the distinguished 2-torsion point is $T = (0, 0)$. We have a 2-isogeny

$$\varphi_2 : \mathcal{E} \to \mathcal{E}/\langle T\rangle \simeq \mathcal{E}' : v^2 + uv = u(u^2 + \sqrt{a}u + b)$$

defined by

$$(x, y) \longmapsto (u, v) \quad \text{where} \quad \begin{cases} u = (x^2 + b^2)/x\,, \\ v = y(u/x) + \sqrt{a}u + \frac{bx+b^2}{x}\,. \end{cases}$$

The induced map on Kummer lines is $(X_P : Z_P) \mapsto ((X_P + bZ_P)^2 : X_P Z_P)$.

Example 4. On the Gaudry–Lubicz model $\mathcal{E} : y^2 + xy = x^3 + b^4$, the distinguished 2-torsion point is $T = (0, b^2)$ and we have a 2-isogeny $\varphi_2 : \mathcal{E} \to \mathcal{E}/\langle T\rangle \simeq \mathcal{E}' : y^2 + xy = x^3 + (\sqrt{b})^4$ defined by

$$(x, y) \longmapsto (u, v) \quad \text{where} \quad \begin{cases} u = (x^2 + b^2)/x\,, \\ v = y(u/x) + b^2(u + 1)/x\,. \end{cases}$$

The induced map on Kummer lines is $(X_P : Z_P) \mapsto ((X_P + bZ_P)^2 : X_P Z_P)$.

Example 5. On the Stam model $\mathcal{E} : y^2 + \frac{1}{b}xy = x^3 + ax^2 + b^2$, the 2-torsion point T is $(0, b)$, and the 2-isogeny $\varphi_2 : \mathcal{E} \to \mathcal{E}' : y^2 + \frac{1}{\sqrt{b}}xy = x^3 + \sqrt{a} + b$ is defined by

$$(x, y) \longmapsto (u, v) \quad \text{where} \quad \begin{cases} u = b(x^2 + 1)/x\,, \\ v = \sqrt{b}\big[y(u/x) + (\sqrt{a} + b)(u/x) + (\sqrt{a}b + 1)/x\big]\,. \end{cases}$$

The induced map on Kummer lines is $(X_P : Z_P) \mapsto (b(X_P + Z_P)^2 : X_P Z_P)$.

6 Odd-degree isogenies of ordinary binary curves

We first recall the classic Vélu formulæ [36], specializing them to characteristic 2 and odd-degree isogenies.

In the following, G is a subgroup of odd degree ℓ of an ordinary elliptic curve $\mathcal{E}/\mathbb{F}_{2^m} : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$. (In practice ℓ is prime, but the results in this section hold for composite odd ℓ.) We choose a subset $S \subset G$ such that $G = S \sqcup (-S) \sqcup \{0\}$, and let $s = \#S = (\ell - 1)/2$.

There exists a separable isogeny

$$\varphi : \mathcal{E} \to \mathcal{E}' : y^2 + a'_1xy + a'_3y = x^3 + a'_2x^2 + a'_4x + a'_6 \cong \mathcal{E}/G$$

with kernel G. Our goal in this section is to give explicit formulæ for rational functions φ_x and φ_y defining a normalized quotient isogeny

$$\varphi : (x, y) \longmapsto (\varphi_x(x), \varphi_y(x, y))$$

and for the coefficients a'_i defining its codomain curve $\mathcal{E}'$.

It is known that the rational functions φ_x and φ_y satisfy

$$\varphi_x(P) = \sum_{Q \in G} x(P+Q) - x(Q) \quad \text{and} \quad \varphi_y(P) = \sum_{Q \in G} y(P+Q) - y(Q)$$

with the convention that $x(P) - x(0_{\mathcal{E}}) = x(P)$ and $y(P) - y(0_{\mathcal{E}}) = y(P)$. In general

$$\varphi_y = cy\varphi_x'(x) + \psi_x(x)$$

for some $c \in \mathbb{F}_{2^m}$ and function ψ_x in $\mathbb{F}_{2^m}(x)$ with

$$2\psi_x(x) = -a_1'\varphi_x(x) - a_3' + c(a_1x + a_3)\varphi_x'(x)\,, \tag{19}$$

(see [13, Thm. 9.7.5]), which in characteristic 2 reduces to

$$\varphi_x'(x) = (a_1'\varphi_x(x) + a_3')/(c(a_1x + a_3))\,. \tag{20}$$

The isogeny is *normalized* if $c = 1$.

Theorem 1 (Vélu). *With the notation above: Let*

$$g_y := a_1x + a_3$$

and set

$$\sigma_1 := \sum_{Q \in S} g_y(Q)\,, \quad \sigma_2(x) := \sum_{Q \in S} \frac{g_y(Q)}{(x + x_Q)^2}\,, \quad \text{and} \quad \sigma_3(x) := \sum_{Q \in S} \frac{g_y^2(Q)}{(x + x_Q)^3}\,.$$

Then there is a normalized isogeny $\varphi : \mathcal{E} \to \mathcal{E}'$ with kernel G where $\mathcal{E}'$ has a-invariants

$$(a_1', a_2', a_3', a_4', a_6') = (a_1, a_2, a_3, a_4 + a_1\sigma_1, a_6 + (a_1^2 + a_3)\sigma_1)$$

and φ is defined by the rational map by $(x, y) \mapsto (\varphi_x(x), \varphi_y(x, y))$ with

$$\varphi_x = x + g_y \cdot \sigma_2(x)$$

and

$$\varphi_y = y + g_y \cdot \sigma_3(x) + \sum_{Q \in S} \Big[\frac{a_1y + a_1g_y + x_Q^2 + a_4}{(x + x_Q)^2}\Big]\,.$$

Proof. These are the formulæ of [36, § 3], specialized to characteristic 2 (and odd isogeny degree), with substantial algebraic simplifications. □

Theorem 1 expresses the Kummer-line map φ_x as sum of quotients via $\sigma_2(x)$, which is well-suited to computations in affine coordinates but not projective ones. Instead, we will use Theorem 2, which is an analogue of Hisil and Costello's formula for Montgomery curves in odd characteristic [9, Thm. 1].

Theorem 2. *With the notation above: If $T \in E(\overline{\mathbb{F}}_{2^m}) \setminus G$, then*

$$\varphi_x(x) - \varphi_x(x_T) = \alpha(x + x_T)\Big[\prod_{Q\in S} \frac{x + x_{Q+T}}{x + x_Q}\Big]^2$$

for some $\alpha \neq 0$ in $\overline{\mathbb{F}}_{2^m}$.

Proof. We adapt the proof of Renes [27, Thm. 1] to characteristic 2. We have

$$\varphi_x(x) = x + \sum_{Q\in S} \frac{g_y(Q)(a_1x + a_3)}{(x + x_Q)^2}.$$

Let $G^* := G \setminus \{0_{\mathcal{E}}\}$, so $G^* = S \sqcup (-S)$, and write

$$v(x) = \prod_{Q\in G^*} (x + x_Q) \quad \text{and} \quad v_R(x) = \prod_{\substack{Q\in G^* \\ Q\neq \pm R}} (x + x_Q) \quad \text{for each } R \in S\,.$$

Now $v_R(x)(x + x_R)^2 = v(x)$ (because $x_R = x_{-R}$), so

$$\varphi_x(x) = \frac{1}{v(x)}\Big[xv(x) + \sum_{Q\in S} g_y(Q)(a_1x + a_3)v_Q(x)\Big] = \frac{w(x)}{v(x)}$$

with $\deg w = \#G$.

Setting $h(x) = w(x)v(x_T) + w(x_T)v(x)$, we have $h(x_T) = 0$, and since φ is invariant under translation by elements of G, we have $h(x_{Q+T}) = 0$ for any $Q \in G$. Therefore

$$(x + x_{Q+T}) \mid h(x) \text{ for all } Q \in G\,.$$

We want to show that

$$(x + x_{Q_1+T})(x + x_{Q_2+T}) \mid h(x) \quad \text{for all} \quad Q_1 \neq Q_2 \in G\,,$$

which then implies that

$$\prod_{Q\in G} (x + x_{Q+T}) \mid h(x)\,.$$

If $x_{Q_1+T} \neq x_{Q_2+T}$ then the claim is straightforward. If $x_{Q_1+T} = x_{Q_2+T}$, then either $Q_1 + T = Q_2 + T$ or $Q_1 + T = -(Q_2 + T)$, so $Q_1 + T = -(Q_2 + T)$ because $Q_1 \neq Q_2$. Now, since Q_1 and Q_2 are kernel elements:

$$[2]\varphi(T) = \varphi(Q_1 + T) + \varphi(Q_2 + T) = \varphi(Q_1 + T) - \varphi(Q_1 + T) = 0_{\mathcal{E}'}\,.$$

Setting $T' = \varphi(T)$, we have $T' \in \mathcal{E}'[2]$, and $T' \neq 0_{\mathcal{E}}$ because $T \notin G$. Hence, T' is the non-trivial 2-torsion point on E', so $a_1'x_{T'} + a_3' = 0$.

Looking at (20), we have $\varphi_x'(x_{Q_1+T}) = 0$ because $\varphi(Q_1 + T) = T'$. Finally, because $h(x) = v(x)v(x_T)(\varphi_x(x) + \varphi_x(x_T))$, we recover $h'(x_{Q_1+T}) = 0$, and thus

$$(x + x_{Q_1+T})^2 = (x + x_{Q_1+T})(x + x_{Q_2+T}) \mid h(x)\,.$$

Because $\deg v < \deg w = \#G$, we must have

$$h(x) = \alpha_0 \prod_{Q \in G} (x + x_{Q+T})$$

for some $\alpha_0 \neq 0$ in $\overline{\mathbb{F}}^*_{2^m}$. Because $\varphi_x(x) = \frac{h(x)}{v(x)v(x_T)} + \varphi_x(x_T)$, we get the result with $\alpha = \frac{\alpha_0}{v(x_T)} \in \overline{\mathbb{F}}^*_{2^m}$. □

Note that the constant α is rational when $T \in \mathcal{E}(\mathbb{F}_{2^m})$. This product formula is the one we will rely on in the sequel.

Remark 6. Theorem 2 gives no information on the function ψ_x. This is because the term relating ψ_x and φ_x in Eq. (19) vanishes in characteristic 2 (thus reducing to Eq. (20), which is what we use in the proof of Theorem 2). Nonetheless, $\psi_x(x)$ satisfies a degree 2 polynomial equation, but it is harder to derive [13, Ex. 9.7.6, Sol. p. 613]. We work on the level of Kummer lines in our applications, so we do not need an expression for ψ_x in any case.

7 Applications

We now apply Theorem 2 to our models. Starting from an ordinary curve $\mathcal{E}$, we know that it is isomorphic to $\mathcal{E}_0$ or $\mathcal{E}_1$, hence we write it as

$$\mathcal{E} : y^2 + xy = x(x^2 + ax + b^2)$$

where $b \neq 0$ and a is either 0 or ε from Definition 1, such that $\mathcal{E} \simeq \mathcal{E}_{\mathrm{Tr}\, a}$. The 2-torsion point is $T = (0, 0)$. Suppose we have a subgroup $G \subset \mathcal{E}$ of odd prime order ℓ; we want to compute the x-map φ_x of an isogeny $\varphi : \mathcal{E} \to \mathcal{E}'$ with kernel G, along with the codomain $\mathcal{E}'$.

Since ℓ is odd, φ preserves 2- and 4-torsion structures; in particular, we can take $\mathcal{E}'$ in the same model as $\mathcal{E}$ with some $b' \neq 0$ but the same a as $\mathcal{E}$: that is,

$$\mathcal{E}' : y^2 + xy = x(x^2 + ax + b'^2)\,.$$

7.1 Enumerating kernel x-coordinates

Given the formula from Theorem 2, we need to choose a subset $S \subset G$ such that $G^* := G \setminus \{0\} = S \sqcup (-S)$, and then compute the x-coordinate of each $Q \in S$. Suppose we have a generator Q_0 of G. We can take $S = \{[i]Q_0 : 1 \leq i \leq s\}$ with $s = \#S = (\ell - 1)/2$. In particular, if $\ell = 3$ then $S = \{Q_0\}$; otherwise, we compute $(X_{[2]Q_0} : Z_{[2]Q_0}) = \texttt{xDBL}((X_{Q_0} : Z_{Q_0}))$. A first naive approach is to compute recursively the remaining $x([k]Q_0)$ using the relation

$$(X_{[k+1]Q_0} : Z_{[k+1]Q_0}) = \texttt{xADD}((X_{[k]Q_0} : Z_{[k]Q_0}), (X_{Q_0} : Z_{Q_0}), (X_{[k-1]Q_0} : Z_{[k-1]Q_0}))$$

The total cost is 1 xDBL and $s-2$ xADDs. In our binary setting, given xDBL is so much cheaper than xADD here, this could be improved by turning half of xADDs into xDBLs when k is even via

$$(X_{[k]Q_0} : Z_{[k]Q_0}) = \texttt{xDBL}((X_{[k/2]Q_0} : Z_{[k/2]Q_0}))\,.$$

The cost is then $\lceil \frac{s-1}{2} \rceil$ xDBLs and $\lfloor \frac{s-1}{2} \rfloor$ xADDs.

For many ℓ, this can be further improved using the techniques of [3, §4], which selects different S to replace xADDs with as many xDBLs as possible. For example, if $\ell = 13$, then we can take $S = \{Q_0, [2]Q_0, [4]Q_0, [8]Q_0 = [-5]Q_0, [16]Q_0 = [-3]Q_0, [32]Q_0 = [-6]Q_0\}$, for which the x-coordinates can be computed using 5 xDBLs (instead of the 3 xDBLs and 2 xADDs suggested above). However, the savings depend heavily on factors like the order of 2 modulo ℓ. Ultimately, the most efficient strategy for enumerating ℓ-isogney kernel x-coordinates must be determined on a case-by-case basis.

Finally, since we use Vélu's formulæ to compute the codomain, normalizing kernel points by computing $x_Q = X_Q/Z_Q$ for every $Q \in S$ can save some multiplications while evaluating. With twisted coordinates, it is better to compute $\widetilde{x}_Q^{-1} = \widetilde{Z}_Q/\widetilde{X}_Q$. In both cases, the cost is $s\mathbf{I}$, which can be reduced to $(3s-3)\mathbf{M}+1\mathbf{I}$ using Montgomery's simultaneous inversion (see e.g. [31, Lem. 1]).

7.2 The isogeny formula

We now apply the formula from Theorem 2:

$$\varphi_x(x) = \alpha(x + x_T)\Big[\prod_{Q\in S} \frac{x + x_{Q+T}}{x + x_Q}\Big]^2 + \varphi_x(x_T)\,.$$

We take $T = (0, 0)$ as the special point: as a point of order 2, it is guaranteed not to be in the (odd) kernel. The order-2 point T on $\mathcal{E}$ is mapped to the order-2 point $T' = (0, 0)$ on $\mathcal{E}'$, so $\varphi_x(0) = \varphi_x(x_T) = x_{T'} = 0$. Similarly, the points of order 4 on $\mathcal{E}$ have x-coordinate b, and are mapped to the points of order 4 on $\mathcal{E}'$, so $\varphi_x(b) = b'$. Finally, we already worked out the translation-by-T morphism: $x_{P+T} = b^2/x_P$ when $P \neq 0_{\mathcal{E}}, T$. The equation for α is then

$$b' = \alpha b\Big[\prod_{Q\in S} \frac{bx_Q + b^2}{bx_Q + x_Q^2}\Big]^2 = \alpha b\Big[\prod_{Q\in S} \frac{b}{x_Q}\Big]^2 \implies \alpha = \frac{b'}{b^\ell}\Big[\prod_{Q\in S} x_Q\Big]^2\,.$$

The expression for φ_x therefore simplifies to

$$\varphi_x(x) = \frac{b'}{b^\ell} x\Big[\prod_{Q\in S} \frac{xx_Q + b^2}{x + x_Q}\Big]^2\,. \tag{21}$$

Therefore, computing b' gives the codomain and the formula for φ_x.

It remains to express these formulæ in projective and twisted-projective coordinates, and to count the cost of evaluating them.

Proposition 3. *Suppose $\mathcal{E} : y^2 + xy = x(x^2 + ax + b^2)$ and $\mathcal{E}' : v^2 + uv = u(u^2 + au + b'^2)$ are connected by an ℓ-isogeny*

$$\varphi : \mathcal{E} \to \mathcal{E}'$$

with kernel G, where ℓ is an odd prime. Let $s = \#S = (\ell - 1)/2$.

1. *On the Kummer lines, $\varphi_x : \mathbb{P}^1 \to \mathbb{P}^1$ is defined by $\varphi_x : (X : Z) \mapsto (\varphi_X(X, Z) : \varphi_Z(X, Z))$ where*

$$\varphi_X(X, Z) = b'X\Big[\prod_{Q\in S}\big(XX_Q + (bZ)(bZ_Q)\big)\Big]^2$$
$$\varphi_Z(X, Z) = bZ\Big[\prod_{Q\in S}\big(X(bZ_Q) + X_Q(bZ)\big)\Big]^2.$$

   We can compute the image of a point under φ_x in $s(5\mathbf{M} + 1\mathbf{m}_0) + 2\mathbf{S} + 2\mathbf{m}_0$, or $5s\mathbf{M} + 2\mathbf{S} + 2\mathbf{m}_0$ if we have precomputed bZ_Q for every $Q \in S$.
2. *If we can precompute $x_Q := \frac{X_Q}{Z_Q}$ for $Q \in S$, then we can replace the formulæ above with the projectively equivalent*

$$\varphi_X(X, Z) = b'X\Big[\prod_{Q\in S}\big(Xx_Q + b(bZ)\big)\Big]^2,$$
$$\varphi_Z(X, Z) = bZ\Big[\prod_{Q\in S}\big(bX + x_Q(bZ)\big)\Big]^2,$$

   which can be evaluated in $4s\mathbf{M} + 2\mathbf{S} + 3\mathbf{m}_0$.

Proof. Starting from Eq. (21), using $x = X/Z$, we have

$$\varphi_x(x) = \frac{b'}{b^\ell}\frac{X}{Z}\Big[\prod_{Q\in S}\frac{XX_Q/(ZZ_Q) + b^2}{X/Z + X_Q/Z_Q}\Big]^2 = \frac{1}{b^{2s}}\frac{b'X}{bZ}\Big[\prod_{Q\in S}\frac{XX_Q + b^2ZZ_Q}{XZ_Q + X_QZ}\Big]^2.$$

Since $\ell = 2s + 1$, we get

$$\varphi_x(x) = \frac{b'X}{bZ}\Big[\prod_{Q\in S}\frac{XX_Q + b^2ZZ_Q}{X(bZ_Q) + X_Q(bZ)}\Big]^2.$$

The formulas for $(\varphi_X : \varphi_Z)$ follow on separating numerators and denominators. Regarding the cost:

- we compute bZ and bZ_Q for every $Q \in S$: $(s + 1)\mathbf{m}_0$;
- each term inside the product: $s(3\mathbf{M})$ (using Eq. 9);
- $(s - 1)$ multiplications on both numerator and denominator: $2(s - 1)\mathbf{M}$;
- remaining operations outside the product (multiplication by b', squares and multiplication by X and Z): $2\mathbf{M} + 2\mathbf{S} + 1\mathbf{m}_0$.

Summing all of these gives the cost of the first part, whether or not we have stored the result of bZ_Q. The formulæ of the second part are clearly equivalent to those of the first, but computing $Xx_Q + b(bZ)$ and $bX + x_Q(bZ)$ requires only $2\mathbf{M}$, rather than the $3\mathbf{M}$ required to compute $XX_Q + (bZ)(bZ_Q)$ and $X(bZ_Q) + X_Q(bZ)$ in the first part. □

We also get an easy analogue for the twisted coordinates $(\widetilde{X} : \widetilde{Z}) = (bZ : X)$ resp. $(b'Z : X)$ on the Kummer lines of $\mathcal{E}$ resp. $\mathcal{E}'$.

Proposition 4. *Using twisted coordinates,*

1. *The Kummer-line map $\varphi_x : \mathbb{P}^1 \to \mathbb{P}^1$ is defined by $(\widetilde{X} : \widetilde{Z}) \mapsto (\widetilde{\varphi}_X : \widetilde{\varphi}_Z)$, where*

$$\widetilde{\varphi}_X(\widetilde{X}, \widetilde{Z}) = \widetilde{X}\Big[\prod_{Q\in S}\big(\widetilde{X}_Q\widetilde{Z} + \widetilde{X}\widetilde{Z}_Q\big)\Big]^2$$
$$\widetilde{\varphi}_Z(\widetilde{X}, \widetilde{Z}) = \widetilde{Z}\Big[\prod_{Q\in S}\big(\widetilde{X}\widetilde{X}_Q + \widetilde{Z}\widetilde{Z}_Q\big)\Big]^2.$$

   The image of a point under this map can be computed in $5s\mathbf{M} + 2\mathbf{S}$.
2. *If we can precompute $\widetilde{x}_Q^{-1} = \frac{\widetilde{Z}_Q}{\widetilde{X}_Q}$ for each $Q \in S$, then we can replace the formulæ above with the projectively equivalent*

$$\widetilde{\varphi}_X(\widetilde{X}, \widetilde{Z}) = \widetilde{X}\Big[\prod_{Q\in S}\big(\widetilde{Z} + \widetilde{X}/\widetilde{x}_Q\big)\Big]^2,$$
$$\widetilde{\varphi}_Z(\widetilde{X}, \widetilde{Z}) = \widetilde{Z}\Big[\prod_{Q\in S}\big(\widetilde{X} + \widetilde{Z}/\widetilde{x}_Q\big)\Big]^2,$$

   which can be evaluated in $4s\mathbf{M} + 2\mathbf{S}$.

Proof. Replacing $(X : bZ)$ in Proposition 3 with $(\widetilde{Z} : \widetilde{X})$ saves computing bZ, multiplying by b', and precomputing the bZ_Q, leading to a cost of $5s\mathbf{M} + 2\mathbf{S}$ for the formulæ in the first part. As in the proof of Proposition 3, the formulæ of the second part are clearly equivalent to those of the first, but we can compute $\widetilde{Z} + \widetilde{X}/\widetilde{x}_Q$ and $\widetilde{X} + \widetilde{Z}/\widetilde{x}_Q$ using only $2\mathbf{M}$ rather than the $3\mathbf{M}$ required to compute $\widetilde{X}_Q\widetilde{Z} + \widetilde{X}\widetilde{Z}_Q$ and $\widetilde{X}\widetilde{X}_Q + \widetilde{Z}\widetilde{Z}_Q$. □

7.3 Computing the codomain

Consider an isogeny φ leaving $\mathcal{E}$. The a-invariants of the codomain $\widetilde{\mathcal{E}}$ are given in Theorem 1, which simplify to

$$\widetilde{a_1} = 1,\ \widetilde{a_2} = a,\ \widetilde{a_3} = 0,\ \widetilde{a_4} = b^2 + v,\ \widetilde{a_6} = v \quad \text{where} \quad v = \sum_{Q\in S} x_Q\,.$$

To get this curve into the right model, we use an isomorphism $\widetilde{\mathcal{E}} \to \mathcal{E}'$ using the formulæ of Proposition 1 with $(u, r, s, t) = (1, 0, 0, \sqrt{v})$, which yields

$$b' = b + \sqrt{v + \sqrt{v}}\,.$$

Computing v is straightforward when we have normalized kernel points; then we only need two square roots to compute the codomain. If we work with twisted coordinates, then we can easily compute

$$\frac{v}{b} = \sum_{Q \in S} \widetilde{x}_Q^{-1}$$

and then recover v by multiplying by b, adding $1\mathbf{m}_0$ to the computation.

8 Implementation and benchmarks

8.1 Experimental setup.

We benchmarked x-evaluation for odd-degree isogenies over $\mathbb{F}_{2^{511}}$, using the base curve $\mathcal{E}$ detailed below. For each prime degree ℓ, we sample a kernel generator $Q_0 \in \mathcal{E}[\ell]$, form the half-kernel set $S = \{Q_0, [2]Q_0, \ldots, [\frac{\ell-1}{2}]Q_0\}$, compute the codomain curve $\mathcal{E}'$, and measure the average time to evaluate the x-map on random inputs. Each reported timing is averaged over 50×32 evaluations (random x-values avoiding poles). Cycle counts are collected on x86_64 using `RDTSCP`. All experiments were run on a 13th Gen Intel Core i7-1365U at up to 5.20 GHz; we compiled the code with GCC 15.2.1 using `-O3`.[2]

Curve details. For the best comparison with prior work, we implemented isogeny computations for a curve isomorphic to Ampe's curve from [2, App. B]. We consider the base curve

$$\mathcal{E}/\mathbb{F}_{2^{511}} : y^2 + xy = x^3 + b^2 x = x(x^2 + b^2)$$

over the binary field

$$\mathbb{F}_{2^{511}} = \mathbb{F}_2[z]/(z^{511} + z^{10} + 1)\,,$$

where b is defined as follows: if $b = \sum_{i=0}^{510} b_i z^i$ with each $b_i \in \{0, 1\}$, then (in hexadecimal)

$$\sum_{i=0}^{510} b_i 2^i = \begin{matrix}\texttt{604E5E6AFF56328CF6FCDCA9AD411B2FC2A6658A6ECE6E5B604FFCDE8ED2BA6F} \\ \texttt{21397DF828D26F295F9D1C7B9FA7BDB1606B85539E0FA85C7BF6B08532582E22}\end{matrix}.$$

(Ampe's curve was presented in the form $y^2 + xy = x^3 + A$; their A is our b^4.)

[2] Our implementation is available in https://anonymous.4open.science/r/submission_sac-1AC6/.

Table 2: Average cycles for one x-evaluation in $\mathbb{F}_{2^{511}}$, averaged over 50×32 calls per ℓ.

ℓ	Vélu (Thm. 1)	Product (Eq. 21)	Prop. 3-aff (1 inv)	Prop. 3-proj (0 inv)	Prop. 4-tw (0 inv)
3	10 393 169	28 879 936	10 543 881	75 216	74 440
5	19 621 248	39 783 668	10 555 526	125 393	123 124
7	32 175 389	50 871 593	10 010 506	160 394	155 171
13	60 861 058	81 195 744	10 945 638	387 269	418 586
17	79 744 703	109 133 308	10 716 132	433 497	383 877
19	93 595 902	107 405 829	10 220 777	418 651	422 548
29	139 227 239	157 480 631	10 439 875	639 577	652 875

Table 3: CRS-action timing over 8 steps, carrying 4 x-values per step. Reported totals include kernel sampling, halfset construction, codomain computation, and x-evaluation.

Variant	Total time (ns)	Time/step (ns)	Total cycles	Cycles/step
CRS-action (Vélu x-eval, Thm. 1)	19,747,964,797	2,468,495,599	53,082,524,644	6,635,315,580
CRS-action (Product x-eval, Eq. 21)	22,167,071,028	2,770,883,878	59,585,080,832	7,448,135,104
CRS-action (Prop. 3-aff: 1 inv)	26,082,877,836	3,260,359,729	70,110,768,710	8,763,846,088
CRS-action (Prop. 3-proj: 0 inv)	17,149,444,126	2,143,680,515	46,097,687,026	5,762,210,878
CRS-action (Prop. 4-tw: 0 inv)	19,371,518,876	2,421,439,859	52,070,637,270	6,508,829,658

The prime factorization of the group order is

$$\begin{aligned}\#\mathcal{E}(\mathbb{F}_{2^{511}}) = {} & 2^3 \cdot 3 \cdot 5 \cdot 7 \cdot 13^2 \cdot 17 \cdot 19 \cdot 29 \cdot 43 \cdot 1951 \cdot 3019 \cdot 32587 \\ & \cdot 3614270784143 \cdot 1830691143712251809111 \\ & \cdot 923192541328873666758617065686505788495707220995298811884 0 \\ & 64950022118301526042300671831047904944 41 .\end{aligned}$$

8.2 Results

Table 2 reports the average cost (in cycles) of evaluating ℓ-isogeny x-maps for $\ell \in \{3, 5, 7, 13, 17, 19, 29\}$. Our baseline affine product implementation is slower than Vélu in this setting, since it performs one inversion per kernel point. In contrast, Propositions 3 and 4 moves inversions out of the kernel loop: the "Prop. 3-aff" variant uses a single inversion per evaluation, while the projective and twisted variants avoid inversions entirely when outputs are carried in projective/twisted form.

In addition to the per-ℓ x-evaluation microbenchmarks above, we report below a higher-level benchmark that mirrors the CRS-action workflow (kernel sampling, halfset construction, codomain computation, and x-evaluation on four carried x-values) over 8 successive steps. The measured totals are shown in Table 3. Since this benchmark uses the CRS-action loop (and its associated degree schedule), its per-call timings are not meant to correspond to a single ℓ row of Table 2.

For this CRS-action benchmark, the affine product implementation runs at 0.64× the speed of Vélu (i.e., slower), while Proposition 3 yields 4.22× for affine

outputs (one inversion), and 24.41× / 20.31× for projective / twisted outputs, respectively (zero inversions during evaluation).

Remark 7. In our implementation, we use a straightforward, portable computation of the binary field $\mathbb{F}_{2^{511}}$, without architecture-specific optimizations (e.g., carryless-multiplication opcodes). We implement inversion with an Itoh–Tsujii addition chain and squaring via a dedicated linear routine; nevertheless, multiplication, square roots, and other linear maps are not micro-architecturally tuned. As a result, the reported timings should be interpreted as reflecting algorithmic differences rather than the peak performance of a fully optimized field-arithmetic backend. Moreover, if the binary field is not fixed, which might be the case of certain applications, the inversion with Itoh–Tsujii addition chain might not be applicable.

9 Conclusion and perspectives

In this paper, we revisited *x-only* evaluation of isogenies on binary elliptic curves, naturally focusing on odd-degree isogenies of ordinary curves (since there is only one supersingular curve, and at most one separable isogeny of degree 2). We obtained multiplicative formulas for isogeny x-maps, extending the approach of Hisil–Costello and Renes to characteristic 2. Specializing to the fast binary model $\mathcal{E} : y^2 + xy = x(x^2 + ax + b^2)$ of Kohel and Pornin, we derived efficient evaluation formulæ in projective and twisted-projective coordinates (Propositions 3 and 4) that avoid inversions during evaluation and allow carrying Kummer points across steps in isogeny chains. Our cost analysis improves on Vélu-style x-evaluation in this setting. Over $\mathbb{F}_{2^{511}}$, microbenchmarks of our reference implementation confirm that, when outputs are kept in projective/twisted form, the inversion-free projective and twisted variants yield speedups over Vélu-style x-evaluation.

9.1 Binary CRS and carried points

A natural next step is to integrate our evaluation formulas into the Couveignes–Rostovtsev–Stolbunov (CRS) group action setting [11, 30]. In CRS-style protocols, each action by a group element is computed as a series of hundreds of isogenies of small degree, where each step *pushes forward* a small set of *carried points* used for kernel sampling in later steps and state propagation. Since this push-forward stage is executed at every step, it is often performance-critical.

So far, CRS protocols have been instantiated using ordinary curves over large-characteristic extension fields [12], and most successfully using supersingular curves over $\mathbb{F}_p$ (CSIDH [8]). But even in CSIDH, the drawback remains the relative slowness of isogeny computations and scalar multiplications, which was a major motivation for Ampe's study of CRS over binary fields [2].

Ampe's results are encouraging, but they suffer from a lack of optimized curve and isogeny arithmetic; our theoretical results fill this gap. Our microbenchmarks over $\mathbb{F}_{2^{511}}$ indicate that avoiding inversions and carrying points in projective/twisted Kummer coordinates can provide substantial speedups in the

CRS "push-forward" stage, although the exact cycle-level gains depend on the efficiency of the underlying field implementation.

Proposition 3 suggests a further concrete optimization for binary CRS: represent carried points entirely in projective (or twisted) Kummer coordinates throughout the group action computation, using our inversion-free x-only evaluation formulas, and deferring dehomogenization to the end (or amortizing it via a single batch inversion). An end-to-end implementation and benchmark of such a "projective-carried-points" CRS walk in characteristic 2, including the interaction with kernel normalization and codomain computation, is left for future work.

References

[1] Marius A. Aardal, Gora Adj, Diego F. Aranha, Andrea Basso, Isaac Andrés Canales Martínez, Jorge Chá vez-Saab, Maria Corte-Real Santos, Pierrick Dartois, Luca De Feo, Max Duparc, Jonathan Komada Eriksen, Tako Boris Fouotsa, Décio Luiz Gazzoni Filho, Basil Hess, David Kohel, Antonin Leroux, Patrick Longa, Luciano Maino, Michael Meyer, Kohei Nakagawa, Hiroshi Onuki, Lorenz Panny, Sikhar Patranabis, Christophe Petit, Giacomo Pope, Krijn Reijnders, Damien Robert, Francisco Rodrí guez-Henríquez, Sina Schaeffler, and Benjamin Wesolowski. *SQIsign*. Tech. rep. National Institute of Standards and Technology, 2025. URL: https://sqisign.org.

[2] Sarah Ampe. "Towards a Speed-up of CRS Using Binary Fields". MA thesis. KU Leuven, 2021.

[3] Gustavo Banegas, Valerie Gilchrist, Anaëlle Le Dévéhat, and Benjamin Smith. "Fast and Frobenius: Rational Isogeny Evaluation over Finite Fields". In: *Progress in Cryptology - LATINCRYPT 2023 - 8th International Conference on Cryptology and Information Security in Latin America , LATINCRYPT 2023, Quito, Ecuador, October 3-6, 2023, Proceedings*. Ed. by Abdelrahaman Aly and Mehdi Tibouchi. Vol. 14168. Lecture Notes in Computer Science. Springer, 2023, pp. 129–148.

[4] Razvan Barbulescu, Damien Robert, and Nicolas Sarkis. *Models of Kummer Lines and Galois Representations*. Cryptology ePrint Archive, Paper 2025/543. 2025.

[5] Daniel J. Bernstein, Peter Birkner, Marc Joye, Tanja Lange, and Christiane Peters. "Twisted Edwards Curves". In: *Progress in Cryptology – AFRICACRYPT 2008*. Ed. by Serge Vaudenay. Springer, 2008, pp. 389–405. ISBN: 978-3-540-68164-9. DOI: 10.1007/978-3-540-68164-9_26.

[6] Daniel J. Bernstein, Luca De Feo, Antonin Leroux, and Benjamin Smith. "Faster computation of isogenies of large prime degree". In: *ANTS XIV*. Ed. by Steven D. Galbraith. The Open Book Series 4. 2020, pp. 39–55. DOI: 10.2140/obs.2020.4.39.

[7] Daniel J. Bernstein, Tanja Lange, and Reza Rezaeian Farashahi. "Binary Edwards Curves". In: *Cryptographic Hardware and Embedded Systems – CHES 2008*. Ed. by Elisabeth Oswald and Pankaj Rohatgi. Vol. 5154.

Springer Berlin Heidelberg, 2008, pp. 244–265. ISBN: 978-3-540-85053-3. DOI: 10.1007/978-3-540-85053-3_16.

[8] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. "CSIDH: An Efficient Post-Quantum Commutative Group Action". In: *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part III*. Ed. by Thomas Peyrin and Steven D. Galbraith. Vol. 11274. Lecture Notes in Computer Science. Springer, 2018, pp. 395–427. DOI: 10.1007/978-3-030-03332-3_15. URL: https://doi.org/10.1007/978-3-030-03332-3%5C_15.

[9] Craig Costello and Hüseyin Hisil. "A Simple and Compact Algorithm for SIDH with Arbitrary Degree Isogenies". In: *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part II*. Ed. by Tsuyoshi Takagi and Thomas Peyrin. Lecture Notes in Computer Science. Springer, 2017, pp. 303–329. DOI: 10.1007/978-3-319-70697-9_11. URL: https://doi.org/10.1007/978-3-319-70697-9%5C_11.

[10] Craig Costello and Benjamin Smith. "Montgomery curves and their arithmetic – The case of large characteristic fields". In: *Journal of Cryptograhic Engineering* 8.3 (2018), pp. 227–240. DOI: 10.1007/S13389-017-0157-6.

[11] Jean-Marc Couveignes. *Hard Homogeneous Spaces*. Cryptology ePrint Archive, Paper 2006/291. 2006.

[12] Luca De Feo, Jean Kieffer, and Benjamin Smith. "Towards Practical Key Exchange from Ordinary Isogeny Graphs". In: *Advances in Cryptology – ASIACRYPT 2018*. Ed. by Thomas Peyrin and Steven Galbraith. Springer International Publishing, 2018, pp. 365–394. ISBN: 978-3-030-03332-3. DOI: 10.1007/978-3-030-03332-3_14.

[13] Steven D. Galbraith. *Mathematics of Public Key Cryptography*. Cambridge University Press, 2012. ISBN: 978-1-107-01392-6. Text references correspond to the 2018 online extended version.

[14] Pierrick Gaudry and David Lubicz. "The Arithmetic of Characteristic 2 Kummer Surfaces and of Elliptic Kummer Lines". In: *Finite Fields and Their Applications* 15.2 (Apr. 1, 2009), pp. 246–260. ISSN: 1071-5797. DOI: 10.1016/j.ffa.2008.12.006.

[15] J. W. P. (James William Peter) Hirschfeld. *Projective Geometries over Finite Fields*. Oxford : Clarendon Press ; New York : Oxford University Press, 1979. 498 pp. ISBN: 978-0-19-853526-3.

[16] Sabyasachi Karati. "Binary Kummer Line". In: *Applied Cryptography and Network Security*. Ed. by Mehdi Tibouchi and XiaoFeng Wang. Springer Nature Switzerland, 2023, pp. 363–393. ISBN: 978-3-031-33488-7. DOI: 10.1007/978-3-031-33488-7_14.

[17] Neal Koblitz. "Constructing Elliptic Curve Cryptosystems in Characteristic 2". In: *Advances in Cryptology – CRYPTO '90*. Ed. by Alfred J. Menezes and

Scott A. Vanstone. Springer, 1991, pp. 156–167. ISBN: 978-3-540-38424-3. DOI: 10.1007/3-540-38424-3_11.

[18] David Kohel. "Efficient Arithmetic on Elliptic Curves in Characteristic 2". In: *Progress in Cryptology - INDOCRYPT 2012*. Ed. by Steven Galbraith and Mridul Nandi. Springer, 2012, pp. 378–398. ISBN: 978-3-642-34931-7. DOI: 10.1007/978-3-642-34931-7_22.

[19] David Kohel. "Endomorphism rings of elliptic curves over finite fields". PhD thesis. University of California at Berkeley, 1996.

[20] Reynald Lercier and David Lubicz. "Counting Points on Elliptic Curves over Finite Fields of Small Characteristic in Quasi Quadratic Time". In: *Advances in Cryptology – EUROCRYPT 2003*. Ed. by Eli Biham. Springer, 2003, pp. 360–373. ISBN: 978-3-540-39200-2. DOI: 10.1007/3-540-39200-9_22.

[21] Julio López and Ricardo Dahab. "Fast Multiplication on Elliptic Curves over GF(2^m) without Precomputation". In: *Cryptographic Hardware and Embedded Systems, First International Workshop, CHES'99, Worcester, MA, USA, August 12-13, 1999, Proceedings*. Ed. by Çetin Kaya Koç and Christof Paar. Vol. 1717. Lecture Notes in Computer Science. Springer, 1999, pp. 316–327. DOI: 10.1007/3-540-48059-5_27. URL: https://doi.org/10.1007/3-540-48059-5%5C_27.

[22] Alfred Menezes. *Elliptic Curve Public Key Cryptosystems*. Springer US, 1993. ISBN: 978-1-4613-6403-0 978-1-4615-3198-2. DOI: 10.1007/978-1-4615-3198-2.

[23] Victor S. Miller. "Use of Elliptic Curves in Cryptography". In: *Advances in Cryptology – CRYPTO '85 Proceedings*. Ed. by Hugh C. Williams. Springer, 1986, pp. 417–426. ISBN: 978-3-540-39799-1. DOI: 10.1007/3-540-39799-X_31.

[24] Peter L. Montgomery. "Speeding the Pollard and Elliptic Curve Methods of Factorization". In: *Mathematics of Computation* 48.177 (1987), pp. 243–264. ISSN: 0025-5718, 1088-6842. DOI: 10.1090/S0025-5718-1987-0866113-7.

[25] Thomaz Oliveira, Julio López, and Francisco Rodríguez-Henríquez. "The Montgomery ladder on binary elliptic curves". In: *Journal of Cryptographic Engineering* 8.3 (2018), pp. 241–258. DOI: 10.1007/s13389-017-0163-8.

[26] Thomas Pornin. *Efficient and Complete Formulas for Binary Curves*. Cryptology ePrint Archive, Paper 2022/1325. 2022. URL: https://eprint.iacr.org/2022/1325.

[27] Joost Renes. "Computing Isogenies Between Montgomery Curves Using the Action of (0, 0)". In: *Post-Quantum Cryptography*. Ed. by Tanja Lange and Rainer Steinwandt. Springer International Publishing, 2018, pp. 229–247. ISBN: 978-3-319-79063-3. DOI: 10.1007/978-3-319-79063-3_11.

[28] Damien Robert and Nicolas Sarkis. "Computing 2-Isogenies between Kummer Lines". In: *IACR Communications in Cryptology* 1.1 (Apr. 9, 2024). ISSN: 3006-5496. DOI: 10.62056/abvua69p1.

[29] Damien Robert and Nicolas Sarkis. "Halving Differential Additions on Kummer Lines". In: *Advances in Cryptology – EUROCRYPT 2025*. Ed. by Serge

Fehr and Pierre-Alain Fouque. Springer Nature Switzerland, 2025, pp. 416–445. ISBN: 978-3-031-91095-1. DOI: 10.1007/978-3-031-91095-1_15.

[30] Alexander Rostovtsev and Anton Stolbunov. *Public-Key Cryptosystem Based on Isogenies*. Cryptology ePrint Archive, Paper 2006/145. 2006.

[31] Hovav Shacham and Dan Boneh. "Improving SSL Handshake Performance via Batching". In: *Topics in Cryptology — CT-RSA 2001*. Ed. by David Naccache. Springer, 2001, pp. 28–43. ISBN: 978-3-540-45353-6. DOI: 10.1007/3-540-45353-9_3.

[32] P.W. Shor. "Algorithms for quantum computation: discrete logarithms and factoring". In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*. 1994, pp. 124–134. DOI: 10.1109/SFCS.1994.365700.

[33] Joseph H. Silverman. *The Arithmetic of Elliptic Curves*. Vol. 106. Graduate Texts in Mathematics. Springer, 2009. ISBN: 978-0-387-09494-6. DOI: 10.1007/978-0-387-09494-6.

[34] The Stacks project authors. *The Stacks project*. 2025. URL: https://stacks.math.columbia.edu.

[35] Martijn Stam. "On Montgomery-Like Representationsfor Elliptic Curves over GF(2^k)". In: *Public Key Cryptography - PKC 2003, 6th International Workshop on Theory and Practice in Public Key Cryptography, Miami, FL, USA, January 6-8, 2003, Proceedings*. Ed. by Yvo Desmedt. Vol. 2567. Lecture Notes in Computer Science. Springer, 2003, pp. 240–253. DOI: 10.1007/3-540-36288-6_18.

[36] Jacques Vélu. "Isogénies Entre Courbes Elliptiques". In: *Comptes-Rendus de l'Académie des Sciences* 273 (1971), A238–A241. English translation by Alexandru Ghitza available here.