Some notes on algorithms for abelian varieties

Damien Robert

March 5, 2024

CONTENTS

INTRODUCTION

These notes are a compilation of various notes I wrote over the year on algorithms for abelian varieties, either as accompanying notes to presentations, or as guiding notes for different projects. They are also available in a standalone form at http://www.normalesup.org/~robert/pro/publications/index.html#notes. The current document will be available at http://www.normalesup.org/~robert/pro/publications/notes/notes_av.pdf.

My HDR [Rob21a], defended in June 2021, provides a much more complete overview of the topics from 2021 and before. Still, the notes might be more convenient for a quick overview of these topics. And the notes post 2021 are obviously not included in the HDR.

For each note, I first give a very brief overview of their content, a link with a public implementation if available, and also refer to accompanying articles (if any). If such articles are available, I would recommend them over the accompanying notes, they will probably be much more polished. Still the notes might be interesting as a quick survey of the content of the articles; and also sometimes I give results in these notes that I could not incorporate in the articles (usually due to space reason).

When I wrote my HDR, I began writing a "book" on the general abstract theory of abelian varieties [Rob21b] (unfortunately I haven't have time to work on it since). These notes are oriented to the practical side of abelian varieties, and can such be seen as a natural companion.

1 COMPUTING CYCLIC ISOGENIES USING REAL MULTIPLICATION (2013-04)

These notes are available at http://www.normalesup.org/~robert/pro/publications/notes/2013-04-Peace-Paris-Cyclic-Is
pdf.

They outline algorithms to compute cyclic isogenies. These algorithms were then worked on with Alina Dudeanu, Dimitar Jetchev and Marius Vuille in [DJRV22; Dud16; Vui20].

The goal was to merge the implementation with AVIsogenies [BCR10], but unfortunately this has not been done yet.

Computing cyclic isogenies using real multiplication

Notes of a talk given for the ANR Peace project

Damien Robert

2013–04–19; Updated 2013–04–23

1 Introduction

This notes are an expanded version of a talk [Rob13] I gave the 11 April 2013 for the PEACE meeting in Paris. Since several people who could not attend have asked for informations about this talk I give here a public version. **A word of warning:** these are preliminary notes so they are bound to have mistakes. More importantly I lack a concrete implementation yet.

1.1 Computing isogenies with maximal isotropic kernel

In [CR11] we gave an algorithm to compute isogenies between abelian varieties. More precisely, let A/k be an abelian variety A/k of dimension g represented by its theta null point of level n (in particular A is polarized). Then given a basis $e_1,\dots,e_g$ of a rational kernel $K\subset A[\ell]$ maximally isotropic for the ℓ-Weil pairing (with ℓ prime to $2n$), we explain how to compute $B=A/K$ (via its theta null point of level n) and how to compute the image of a point $x\in A(\overline{k})$ via the isogeny $f:A\to B$. This can be seen as a generalisation of the well known Vélu's formulas [Vél71] to compute isogenies between abelian varieties.

This algorithm needs a polynomial (in the size of the kernel K) number of operations in the field where the geometric points of K live. Actually, the article [CR11] focus on the case of dimension $g=2$, because in this case every (generic) abelian variety is a Jacobian of an hyperelliptic curve, and we explain how to use Thomae's formulas to convert between the Mumford representation and the theta representation (see also [Wam99]).

More details on this algorithm are also given in [Cos11] (using analytic theta functions), and in [Rob10] (using algebraic theta functions). The algorithm given in [CR11] builds on result from [FLR11; LR12b] by applying a result from [Koi76] (in the analytic setting) and [Kem89] (in the algebraic setting).

The above algorithm was implemented in [BCR10a] to compute isogenies between abelian varieties of dimension 2 over finite fields. Note that when one use the theory of analytic theta functions, to extend the results to varieties over a finite field, one need to assume that they are ordinary so that a lift to characteristic zero can be taken. The advantage of algebraic theta functions is that the resulting theory will work over any algebraically closed field of characteristic prime to the level n. Since $n=2$ or $n=4$ this handle all fields of odd characteristics. For an ordinary abelian variety over $\mathbb{F}_{2^n}$, one can lift to characteristic zero, but the formulas from the isogeny algorithm have bad reduction in this case, so we need to make a change of variable. The resulting algorithm to compute isogenies in characteristic two is described in [BCR10b] (for the dimension 2 case).

The condition ℓ prime to $2n$ is purely technical, we explain in [Rob10] how to compute an isogeny when this is not the case (in this case we need more than juste the geometric points of the kernel, we will see why in Section 2).

Finally, an improvement of this algorithm so that only operations over the field of definition of the kernel K are needed (provided we have the equations of K) is given in [Rob12] (in collaboration with David Lubicz).

1.2 The case of cyclic isogenies

At the end of [CR11], we concluded that it would be worthwhile to investigate the case of isogenies with cyclic kernel; they are needed to have a full description of the isogeny graph (otherwise we don't even have a connected

subgraph), which has many applications: [LR12a]... The problem here is that the pullback of a line bundle by a cyclic kernel is not as easy to describe algebraically as when the kernel is maximally isotropic.

It is easier to explain why if we use the theory of complex multiplication [Shi98]. Let K be a (primitive) CM field of degree $2g$ (a totally imaginary quadratic extension of a totally real field K_0). Then the moduli space of abelian varieties with complex multiplication by O_K is a torsor under the Shimura class group

$$\mathfrak{C} = \{(I,\rho) \mid I \text{ a fractional } O_K\text{-ideal with } I\bar{I} = (\rho),\ \rho \in K^+ \text{ totally positive}\}/K^*$$

(In particular, it is of dimension 0.)

If $(A, \mathscr{L}_0)$ is a principally polarized abelian variety of dimension g with CM by O_K, the element $(I,\rho) \in \mathfrak{C}$ acts on A in the following way: I give the kernel K of the corresponding isogeny on A, and ρ explain the action on the polarization. I corresponds to a maximal isotropic kernel (for the Weil pairing on $\mathscr{L}_0^\ell$) iff I is of relative norm ℓ. In this case the element (I,ℓ) give an isogeny between the polarized abelian variety $(A, \mathscr{L}_0^\ell)$ and $(B, \mathscr{M}_0)$ (where $\mathscr{M}_0$ is a principal polarization), so the action of ℓ on the polarization is easy to describe. For a general element (I,ρ), one would need to understand what the polarization "$\mathscr{L}_0^\rho$" such that we have an isogeny $(A, \mathscr{L}_0^\rho)$ and $(B, \mathscr{M}_0)$ of polarized abelian varieties would mean. Note that $\mathscr{L}_0^\rho$ is not isomorphic to $\rho^*\mathscr{L}_0$ (Think about the case $\rho = \ell$ and $\mathscr{L}_0$ symmetric where $\rho^*\mathscr{L}_0 = \mathscr{L}_0^{\ell^2} \neq \mathscr{L}_0^\ell$).

When $\rho = \ell$, one could compute an isogeny (with maximal isotropic kernel for $\mathscr{L}_0^\ell$) the following way: find a matrix $F \in \mathrm{Mat}_r(\mathbb{Z})$ such that ${}^tFF = \ell\,\mathrm{Id}$. Then the Koizumi-Kempf formula applied to F give a link between the theta functions of level ℓn on $\mathscr{L}^\ell$ (where $\mathscr{L} = \mathscr{L}_0^n$) and the theta functions of level n on $\mathscr{L}$, we will call this "changing the level" or the "level formulas". (Basically we just have to apply the isogeny theorem on the isogeny $F : A^r \to A^r$ given composant by composant by the matrix F. Here A^r is given the product polarization $\mathscr{L} \star \ldots \star \mathscr{L}$, so the isogeny theorem give relations between products of r theta functions on A.) Then once we are in $(A, \mathscr{L}^\ell)$ we can just apply the isogeny theorem to get into $(B, \mathscr{M})$ ($\mathscr{M} = \mathscr{M}_0^n$). In [CR11] we do things the other way around because we get a more efficient algorithm this way, we will explain why latter.

In the case of complex multiplication, one could try to adopt a similar strategy for a cyclic isogeny coming from the action of (I,ρ): find a matrix $F \in \mathrm{Mat}_r(O_K)$ such that ${}^tFF = \rho\,\mathrm{Id}$ and apply a Koizumi like formula to get from $(A, \mathscr{L})$ to $(A, \mathscr{L}^\rho)$. We have two problem here: the Koizumi formula comes from the isogeny formula on A^r, but when F is not an integral matrix, there is no reason that F respect the underlying symplectic decomposition, so we may not apply the isogeny theorem. The second problem, is that even if it does, to compute the corresponding change of level, we need a way to compute the action of elements of O_K on affine lifts uniformly. For an action of $\gamma \in \mathbb{Z}$ we know how to do it using differential additions, but it is not clear how to do that for a more general γ. If γ itself correspond to an isogeny with maximal isotropic kernel, then one solution is to use [CR11], because the isogeny algorithm given here actually work with affine coordinates (this is clear given the way we keep track of the projective factors), so it would be doable but would need branching isogeny computations inside the level formula of our current cyclic isogeny computation. All in all this seemed like a cumbersome computation, and it only guides us in the case of fixed CM, whereas I was interested in moving vertically in the isogeny graph using cyclic isogenies.

In November 2011, Dimitar Jetchev contacted me about the possibility of computing cyclic isogenies in dimension 2, and this is basically the response I gave: that in the restricted case of known CM and horizontal isogeny, it should theoretically be feasible but rather cumbersome.

1.3 Real multiplication to the rescue!

In July 2012, while I was visiting Microsoft Research, I discussed with Sorina Ionica who showed me wonderful graphs of cyclic isogenies between abelian varieties having the same real multiplication (RM) in dimension 2. These graphs were obtained in collaboration with Emmanuel Thomé, following an idea from John Boxall to use real multiplication to compute isogenies.

While Sorina and Thomé obtained their graphs by working over $\mathbb{C}$ (and with lattices coming from the Hilbert space $\mathfrak{H}_1^g$ via the real multiplication O_{K_0}), this discussion made clear that the case of computing the action of ρ on the polarization (in order to compute a cyclic isogeny) was much better than I thought.

Indeed, it is clear from the definition of the Shimura class group that ρ is a totally positive element in K_0. It is well known that every such element is a sum of squares, and it is also well known how from such a sum of

squares one can use Clifford's algebra to compute a matrix F such that $t_F F = \rho\,\mathrm{Id}$. The important part here is that $F \in \mathrm{Mat}_r(O_{K_0})$ is composed of totally real elements. This has two important consequences, first since complex conjugation on an ideal $I \subset K$ correspond to the dual isogeny, an element $\gamma \in K_0$ commutes with the Rosatti involution. In particular, the action of the elements of K_0 on $\mathbb{C}^g$ is given by symmetric matrices for the hermitian form H associated to the principal polarization on $(A, \mathscr{L}_0)$. In particular they are all codiagonalisable, so it is immediate that the matrix F respect the symplectic decomposition and we can apply the isogeny theorem to obtain Koizumi-like formulas. Secondly, computing the action of an element in K_0 on affine points of the abelian variety is much easier than for a general element in K as we will see.

Independently, Alina Dudeanu and Dimitar Jetchev have also been working on obtaining a Koizumi-like formula in the analytic setting using real multiplication.

This notes are heavily indebted to helpful discussions with John Boxall and Sorina Ionica; and even more importantly to my on-going collaboration with David Lubicz in the use of algebraic theta functions for cryptographic applications. We will assume known the standard results of analytic theta functions [Igu72; Mum83; Mum84; Mum91; BL04] and algebraic theta functions [Mum66; Mum67a; Mum67b; Mum70]. We use the standard acronyms ppav for principally polarized abelian variety and pav for polarized abelian variety. We will also always assume that the line bundles are symmetric.

2 Symmetric theta structures and the isogeny theorem

Let A be an abelian variety of dimension g defined over an algebraically closed field $\overline{k}$. Let $\mathscr{L}_0$ be a symmetric ample line bundle of degree one on A, $\mathscr{L}_0$ defines a principal polarization: $A \to \hat{A}$. If n is even $\mathscr{L} = \mathscr{L}_0^n$ is then totally symmetric, and the kernel $K(\mathscr{L})$ of the polarization associated to $\mathscr{L}$ is $A[n]$.

From now on, we assume that n is prime to the characteristic of k, so that $\mathscr{L}$ defines a separable polarisation. Since $\mathscr{L}$ is totally symmetric, there exist a symmetric theta structure on the theta group $G(\mathscr{L})$. Fixing such a structure fix a unique projective basis of theta functions [Mum66] that we call theta functions of level n. Note: the theta structure induces an isomorphism between the symplectic spaces $Z(\overline{n}) \times \hat{Z}(\overline{n})$ and $K(\mathscr{L}) = A[n]$ where $Z(\overline{n}) = (\mathbb{Z}/n\mathbb{Z})^g$ and $\hat{Z}(\overline{n})$ is the Cartier dual of $Z(\overline{n})$. We note $K(\mathscr{L}) = K_1(\mathscr{L}) \oplus K_2(\mathscr{L})$ where $K_1(\mathscr{L})$ corresponds to $Z(\overline{n})$ and $K_2(\mathscr{L})$ to $\hat{Z}(\overline{n})$. Usually the canonical basis of the theta functions of level n are indexed by $i \in Z(\overline{n})$, but in these notes we will index them by $i \in K_1(\mathscr{L})$ which permit us to not track explicitly the isomorphism between $Z(\overline{n})$ and $K_1(\mathscr{L})$.

If $n > 2$ then the theta functions of level n give a projective embedding of A into $\mathbb{P}^{n^g-1}_{\overline{k}}$, while if $n = 2$ we only get an embedding of the Kummer variety $A/\pm 1$ (the $n = 2$ case assume that A is absolutely simple, see [BL04]). Under a generic condition (the even theta null coordinates are non zero), this embedding of the Kummer variety is actually projectively normal (see [Koi76]).

Theorem 2.1 :
The symmetric theta structure on $G(\mathscr{L})$ is uniquely determined by a choice of symplectic basis $(e_1, \dots e_g, e'_1, \dots e'_g)$ on $A[n]$ and a choice of symplectic basis $(f_1, \dots f_g, f'_1, \dots f'_g)$ on $A[2n]$ such that $e_i = 2f_i, e'_i = 2f'_i$. (Here symplectic mean for the commutator pairing $e_{\mathscr{L}}$ and $e_{\mathscr{L}^2}$ respectively).

Moreover, changing these symplectic basis do not change the resulting symmetric theta structure if and only if

- *The symplectic basis of $A[n]$ is left invariant;*
- *The f_i are replaced by points $f_i + t_i$ with $t_i \in A[2]$ such that $e_{\mathscr{L}}(e_i, t_i) = 1$.*

In particular, fixing a symplectic basis of $A[n]$ and a symplectic decomposition $A[2n] = A_1[2n] \oplus A_2[2n]$ of the $2n$-torsion into a sum of maximal isotropic subspaces is enough (and even stronger) to fix the symmetric theta structure.

Proof : This is implicit in [Mum66, Section 3]. A symmetric theta structure comes from an isomorphism between the Heisenberg group and the theta group that commutes with the action of $[-1]$. It induces an isomorphism between the symplectic spaces $Z(\overline{n}) \times \hat{Z}(\overline{n})$ and $K(\mathscr{L}) = A[n]$ and hence fix a symplectic basis of the n-torsion.

Conversely, having fixed a symplectic basis of the n-torsion, since $\mathcal{L}$ is totally symmetric, there is always a symmetric theta structure respecting this symplectic basis. Such a choice of a symmetric theta structure can be seen as a choice of a symmetric element above each of the element of the basis $(e_1,\dots e'_g)$; since there is only two symmetric elements $\pm g_i$ above each e_i a symmetric theta structure above the symplectic basis can be seen as a choice of sign for each element of the basis.

If $g_i \in G(\mathcal{L}^2)$ is a symmetric element of the theta group above a point f_i such that $e_i = 2f_i$, then $(g_i)^2$ determines a symmetric element of the theta group above e_i that uniquely depends on the choice of f_i (since the other symmetric element above f_i is $-g_i$ which gives rise to $(-g_i)^2 = (g_i)^2$ above e_i. Via the transfer map δ_2 from [Mum66], we see how the choices of the f_i above the e_i are enough to determine the symmetric theta structure on $G(\mathcal{L})$.

It is a straightforward verification to see that replacing f_i by $f_i + t_i$ where t_i is a point of 2-torsion involve replacing $(g_i)^2$ by $e_{\mathcal{L}^2}(f_i,t_i)(g_i)^2$ which concludes the proof.

(One could also replace the application δ_2 by the isogeny $[2]$ which would involve working in $G(\mathcal{L}^4)$, as in [Kem89].) ■

Of course Theorem 2.1 also work for any totally symmetric line bundle $\mathcal{L}$ on A, defining a polarization of type $\delta = (\delta_1,\dots,\delta_g)$. The idea is that if $\mathcal{L} = \mathcal{L}_0^n$ (say with $n=2$ or $n=4$), $\mathcal{L}^\ell$ is of type $(\ell n, \ell n)$ and allows to compute isogenies with maximal isotropic kernels, but for a cyclic isogeny we need a polarisation of type $(n,\ell n)$ (like the type of $\mathcal{L}^\rho$ from Section 1.3).

Theorem 2.2 :
Let $f : (A,\mathcal{L}) \to (B,\mathcal{M})$ be an isogeny between pav, with $\mathcal{L}$ totally symmetric. Then $K = \operatorname{Ker} f$ is isotropic in $K(\mathcal{L})$ for the commutator pairing $e_{\mathcal{L}}$, and $K(\mathcal{M}) \simeq K^\perp/K$.

Assume that we have a symmetric theta structure on $G(\mathcal{L})$ coming from a symplectic basis (f_i,f'_i) on $K(\mathcal{L}^2)$. Assume that K is compatible with the induced symplectic decomposition $K(\mathcal{L}) = K_1(\mathcal{L}) \oplus K_2(\mathcal{L})$ into maximal isotropic subspaces in the sense that $K = K_1 \oplus K_2$ where $K_i = K_i(\mathcal{L}) \bigcap K$. In this case $K(\mathcal{M}) \simeq K^{2,\perp}/K_1 \oplus K^{1,\perp}/K_2$ where $K^{2,\perp} = K_2^\perp \bigcap K_1(\mathcal{L})$ and $K^{1,\perp} = K_1^\perp \bigcap K_2(\mathcal{L})$

Let $\widetilde{K}$ be the level subgroup above K induced by this theta structure; the corresponding descent data give a line bundle $\mathcal{M}'$ algebraically equivalent to $\mathcal{M}$. For simplicity we assume here that $K \subset 2K(\mathcal{L})$ (or equivalently that $A[2] \subset K^\perp$), so that $\mathcal{M}'$ is the unique totally symmetric line bundle equivalent to $\mathcal{M}$. (The isogeny theorem is valid in a more general setting, but we will only need this case in the following).

We can define a symmetric theta structure on $\mathcal{M}'$ as follow: from the symplectic basis of $K(\mathcal{L}^2)$ one derives a "canonical" basis $(g_1,\dots,g'_g)$ of $[2]^{-1}K^\perp$. Pushing this basis via the isogeny f gives a symplectic basis on $K(\mathcal{M}'^2)$, which determines the symmetric theta structure on $\mathcal{M}'$. It is easy to see that by construction, it is compatible with the theta structure on $\mathcal{L}$.

We can then apply the isogeny theorem: there exist λ such that for all $i \in K_1(\mathcal{M}')$

$$\vartheta_i^{\mathcal{M}'} = \lambda \sum_{j \in K_1(\mathcal{L}) | f(j) = i} \vartheta_j^{\mathcal{L}}.$$

Proof : This is [Mum66, Section 1]. The version stated here is from [Kem89]. See also [Rob10, Chapter 3–4] for a summary. ■

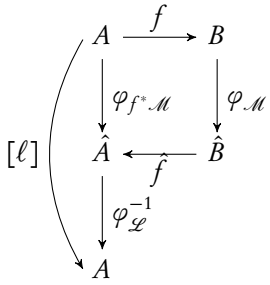
3 Computing isogenies with maximal isotropic kernel

In this section we review the algorithm of [CR11]. This is because we will see that the tools used to compute cyclic isogenies are extremely similar, and also because we will need to be able to compute maximally isotropic isogenies in order to compute cyclic isogenies.

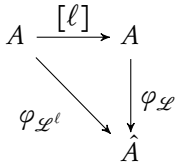
Let $(A,\mathcal{L}_0)$ be a ppav, and K a maximal isotropic kernel for $\mathcal{L}_0^\ell$. Let n be even and $\mathcal{L} = \mathcal{L}_0^n$. Assume that we have a principal polarization $\mathcal{M}_0$ on $B = A/K$, and let $\mathcal{M} = \mathcal{M}_0^n$. For simplicity we assume here that ℓ is prime to $2n$. We note $\Phi_{\mathcal{L}} : A \to \hat{A},\ x \mapsto t_x^*\mathcal{L} \otimes \mathcal{L}^{-1}$ the polarization associated to $\mathcal{L}$.

To have an algorithm for the isogeny $f : A \to B$ mean that we want to find relations between theta functions of level n on A (for $\mathscr{L}$) and theta functions of level n on B (for $\mathscr{M}$).

First we need to have some sort of compatibility between $\mathscr{L}$ and $\mathscr{M}$. More exactly, we want the following diagram to commute:



It is easy to see that since we have the following diagram:



this is the case iff $\mathscr{L}^\ell = f^*\mathscr{M}$.

Now we have two tools. The Koizumi formula explain the relations between the theta functions of level ℓn for $\mathscr{L}^\ell$ and the theta functions of level n for $\mathscr{L}$.

Concretely, assume given a symmetric theta structure on $\mathscr{L}^\ell$, by Theorem 2.1 this induces a symmetric theta structure on $\mathscr{L}$. Let $F \in \mathrm{Mat}_r(\mathbb{Z})$ be a matrix such that ${}^tFF = \ell\,\mathrm{Id}$, and note also F the isogeny $A^r \to A^r$ induced by F. (In practice $r = 2$ when ℓ is a sum of two squares, and $r = 4$ otherwise). The theta structures on $\mathscr{L}$ and $\mathscr{L}^\ell$ induce product theta structures on $\mathscr{L} \star \ldots \mathscr{L}$ and $\mathscr{L}^\ell \star \ldots \mathscr{L}^\ell$. In this setting, Theorem 2.2 gives us

Proposition 3.1 :
Let $(i_1, \ldots, i_r) \in K_1(\mathscr{L})^r$. Let $x = (x_1, \ldots, x_r)$ be a geometric point of A^r and let $y = Fx$. Then (up to a constant λ)

$$\vartheta^{\mathscr{L}}_{i_1}(y_1) \cdot \ldots \cdot \vartheta^{\mathscr{L}}_{i_r}(y_r) = \lambda \sum_{\substack{(j_1,\ldots,j_r) \in K_1(\mathscr{L}^\ell) \\ F(j_1,\ldots,j_r) = (i_1,\ldots,i_r)}} \vartheta^{\mathscr{L}^\ell}_{j_1}(x_1) \cdot \ldots \cdot \vartheta^{\mathscr{L}^\ell}_{j_r}(x_r).$$

Proof : From the theorem of the square we have that $F^*(\mathscr{L} \star \mathscr{L} \star \ldots) = \mathscr{L}^\ell \star \mathscr{L}^\ell \star \ldots$. The rest is immediate from Theorem 2.2. ■

The isogeny formula explain the relations between the theta functions for $\mathscr{L}^\ell$ on A and the theta functions for $\mathscr{M}$ on B.

Proposition 3.2 :
Assume that the symmetric theta structure on $\mathscr{L}^\ell$ is such that $K \subset K_2(\mathscr{L}^\ell)$ (this is always possible). Then the symmetric theta structure on $\mathscr{L}^\ell$ induces a symmetric theta structure on $\mathscr{M}$ by Theorem 2.2 (this may require to replace $\mathscr{M}$ by an equivalent line bundle).

Let $f : A \to B$ be the isogeny of kernel K, and x a geometric point in A. Fix $i \in K_1(\mathscr{M})$, and let $j \in K_1(\mathscr{L}^\ell)$ be the unique preimage of i by f that is in $K_1(\mathscr{L}^\ell)$. We have (up to a constant λ)

$$\vartheta^{\mathscr{M}}_i(f(x)) = \lambda \vartheta^{\mathscr{L}^\ell}_j(x)$$

Proof : Immediate by Theorem 2.2. ■

Example 3.3 :
Let $A = \mathbb{C}^g/(\mathbb{Z}^g + \Omega\mathbb{Z}^g)$ be a ppav over $\mathbb{C}$ with $\Omega \in \mathfrak{H}_g$. The polarization associated to Ω correspond to an hermitian form H_0 on $\mathbb{C}^g$. More generally, a polarization comes from an hermitian form H on $\mathbb{C}^g$ such that $H(ix, iy) = H(x, y)$ and $H(\Lambda, \Lambda) \subset \mathbb{Z}$ where $\Lambda = \mathbb{Z}^g + \Omega\mathbb{Z}^g$.

An isogeny correspond to a matrix M acting on $\mathbb{C}^g$, and the dual isogeny correspond to $H_0(M\cdot, \cdot)$ acting on $\hat{A} \simeq \mathrm{Hom}_{\overline{\mathbb{C}}}(\mathbb{C}^g, \mathbb{C})$. Pulling back the dual isogeny via the principal polarization, we get that it acts on $\mathbb{C}^g$ by $M^* = {}^t\overline{M}$. (We see that we recover the action by F on $\mathscr{L} \star \mathscr{L} \star \dots$ from Proposition 3.1).

A basis of level n theta function corresponding to $H = nH_0$ (and the characteristic $c = 0$ in the sense of [BL04]) is given by $(\vartheta \left[\begin{smallmatrix} 0 \\ b \end{smallmatrix}\right](\cdot, \Omega/n)_{b \in Z(\overline{n})}$ where

$$\vartheta \left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right](z, \Omega) = \sum_{n \in \mathbb{Z}^g} e^{\pi i\, {}^t(n+a)\Omega(n+a) + 2\pi i\, {}^t(n+a)(z+b)}.$$

Up to an action of the symplectic group $\mathrm{Sp}_{2g}(\mathbb{Z})$ we can assume that the kernel K corresponds to $\frac{1}{\ell}\Omega\mathbb{Z}^g/\Omega\mathbb{Z}^g$ so that the isogenous abelian variety is $B = \mathbb{C}^g/(\mathbb{Z}^g + \frac{\Omega}{\ell}\mathbb{Z}^g)$.

Comparing the basis of theta functions of level n on B

$$(\vartheta \left[\begin{smallmatrix} 0 \\ b \end{smallmatrix}\right](\cdot, \frac{\Omega}{\ell}/n))_{b \in Z(\overline{n})}$$

and the basis of theta functions of level $n\ell$ on A

$$(\vartheta \left[\begin{smallmatrix} 0 \\ b \end{smallmatrix}\right](\cdot, \Omega/\ell n))_{b \in Z(\overline{\ell n})}$$

immediately give Proposition 3.2.

Now the natural thing to compute the isogeny $A \to B$ would be to combine Propositions 3.1 and 3.2: inverse the formulas from Proposition 3.1 to go from theta coordinates on $\mathscr{L}$ to theta coordinates on $\mathscr{L}^\ell$ and then apply Proposition 3.1 on $\mathscr{L}^\ell$.

Inversing the level formula could be done as follow: first try to find a theta null point of level ℓn associated to a symmetric theta structure on $G(\mathscr{L}^\ell)$ compatible with the one on $G(\mathscr{L})$. Since we know how the moduli space of theta null point of a certain level look like (by [Mum67a] it is given by Riemann's relations + the symmetries) this can be done by a Gröbner basis algorithm. Since the fiber is finite, we are in a favorable case for Gröbner computations. Then once we have fixed a theta null point of level ℓn in the fiber, we can lift a geometric point x on A given by level n theta coordinates to level ℓn coordinates. This can also be done by a Gröbner basis algorithm since the projective equations of A embedded by theta functions is described in [Mum66] (and only need the coordinates of the theta null point).

In fact, in [FLR11] we inverse the isogeny formula from Proposition 3.2 instead. This is because it is simpler, so it allows to speed-up the Gröbner basis computation related by using the extra information we have about the system (in [FLR11] we only care about lifting the theta null point since we were only interested in describing some modular correspondances). In other words, rather than looking at the isogeny $f : (A, \mathscr{L}^\ell) \to (B, \mathscr{M})$, we look at the contragredient isogeny $\widetilde{f} : B \to A$. (This whole part and what follows is because Ben Smith complained during the talk that we think with "arrows reversed", this is to try to justify why it is a good idea in our situation!)

Still we would like to have an algorithm that does not need Gröbner basis. We note here that both Proposition 3.1 and Proposition 3.2 loose information (they go to a lower level), but only in a finite way (the associated fibers are finites). Theorem 2.1 allow us to keep track of exactly which information is lost. This suggest the following strategy: work on $(A, \mathscr{L})$ directly to recover the extra information needed to lift to level ℓn.

For instance, if we suppose that ℓ is prime to level $2n$, then it is clear from Theorem 2.1 that the choice of a compatible symmetric theta structure on $G(\mathscr{L}^\ell)$ is exactly the choice of a symplectic basis of $A[\ell]$ (we assume here that $\mu_\ell \subset k$). But since K is a maximal isotropic subgroup of the ℓ-torsion, this is the same as a choice of a basis $(e_1, \dots, e_g)$ of K and a supplementary isotropic subgroup of K in $A[\ell]$. This explain the technical condition ℓ prime to $2n$ of Section 1.1; for the general case we need to find a (compatible) symplectic basis of the full $A[2\ell n]$ torsion.

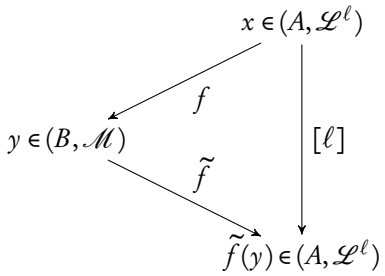
Now let's think "with arrow reversed", and let $K' = f(A[\ell])$ be the kernel of the contragredient isogeny $\widetilde{f} : B \to A$; from K' we want to compute $\widetilde{f}$ algorithmically.

Starting from theta functions of level n on B (from $\mathscr{M}$), we then want to go to theta functions of level ℓn on A (from $\mathscr{L}^\ell$). But the exact same information as before is also enough to fix a symmetric theta structure on $G(\mathscr{L}^\ell)$. Namely, fix a basis of the maximal isotropic group $K' \subset B[\ell]$ and a decomposition $B[\ell] = B_1[\ell] \oplus B_2[\ell]$ with $B_1[\ell] = K'$. This determines a full symplectic basis of the ℓ-torsion. The decomposition of $B[\ell]$ fixes a decomposition of $B[\ell^2]$ and thus a decomposition of $A[\ell]$ via $\widetilde{f}$, and the image of the basis of $B_2[\ell]$ give a basis of $K = A_2[\ell]$.

Concretely, let's look at an example with $g = 1$, $n = 2$ and $\ell = 3$. Then from Proposition 3.2 we readily see that the isogeny f is given by $(x_0, \dots x_5) \mapsto (x_0, x_3)$. Moreover by definition of a theta structure of level n, we can compute the action by translation by any point of n-torsion. In our situation, we are on level ℓn on A and we have a decomposition $A[\ell] = A_1[\ell] \oplus A_2[\ell]$ with $A_2[\ell] = K$. The isomorphism $\mathrm{Z}(\overline{\ell n}) \to A_1[\ell n]$ give us that $A_1[\ell]$ is generated by a point of 3-torsion T such that $(x+T)_i = (x)_{i+2}$ for $i \in \mathrm{Z}(\overline{\ell n})$ (2 being of 3-torsion in $\mathbb{Z}/6\mathbb{Z}$). Then the kernel K' of the contragredient isogeny $\widetilde{f}$ is generated by $f(T)$. We have $f(x+T) = (x_2, x_5)$ and $f(x+2T) = (x_4, x_1)$. We see that we could recover the coordinates of x from the knownledge of $f(x)$ and $f(T)$ if we were able to take "compatible" affine lifts of $f(x)$, $f(x)+f(T)$ and $f(x)+2f(T)$. But this is exactly what the theory of differential addition allow us to do as we explain in [LR12b].

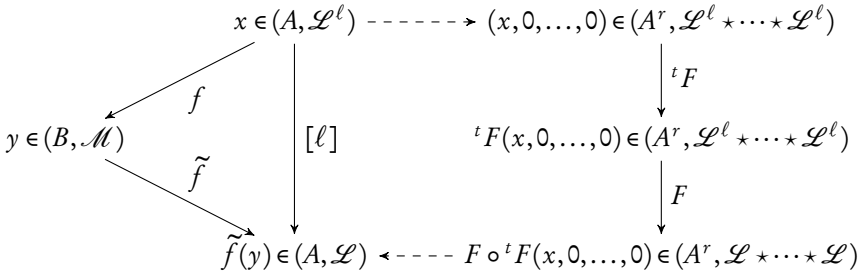
Of course, a similar method applies to go from $(A, \mathscr{L})$ to $(A, \mathscr{L}^\ell)$ by taking uniform affine lifts of points of ℓ-torsion given by their level n theta coordinates. More details are given in [Rob10; Cos11]. So we don't really need to work with "arrow reversed", but in practice it is easier to do so; from a theta null point of level ℓn on A we readily get points of ℓ-torsion in level n on B, but it is a bit more complicated to get points of ℓ-torsion in level n on A. Once again, this come from the difference between the simplicity of the equation in Proposition 3.2 compared to Proposition 3.1.

Now we are almost finished describing the isogeny algorithm. By definition of the contragredient isogeny, the following diagram commutes:



As mentioned, in [LR12b] we explain how to compute from y a point x such that $f(x) = y$. There is some ℓ-root involved, other choices of the root corresponds to different preimages (the preimage does not matter because we multiply it by $[\ell]$ afterwards).

Now $\widetilde{f}(y) = [\ell]x$. We are not quite finished because here $\widetilde{f}(y)$ is given by level ℓn theta functions. So we use the following diagram



Here the computation of ${}^tF$ is done in $\mathscr{L}^\ell$ while we use Proposition 3.1 to compute the action of F in order to go back to level n.

Now fix a basis of K'. There is some ℓ-roots involved for lifting the theta null point of $(B,\mathscr{M})$ to $(A,\mathscr{L}^\ell)$ which correspond to different choices of a supplementary of K' in $B[\ell]$. Now of course these choices does not affect the end result of the computation of $\widetilde{f}(y)\in(A,\mathscr{L})$. In other words, rather than going up on $(A,\mathscr{L}^\ell)$ and then down in $(A,\mathscr{L})$ we only need to have enough informations from $(A,\mathscr{L}^\ell)$ in order to be able to go down to $(A,\mathscr{L})$. We explain how to do that in [CR11], where we get the following

Proposition 3.4 :
Let $(B,\mathscr{M}_0)$ be a ppav with a symmetric theta structure on $G(\mathscr{M})$ where $\mathscr{M}=\mathscr{M}_0^n$ is of level n even. Let $K'\subset B[\ell]$ be a maximal isotropic subgroup and $\widetilde{f}:B\to A=B/K'$ be the associated isogeny. Assume that ℓ is prime to $2n$; then the theta structure on $G(\mathscr{M})$ induces a unique polarization $\mathscr{L}$ of level n on A and a unique compatible symmetric theta structure on $G(\mathscr{L})$. Let $F\in\mathrm{Mat}_r(\mathbb{Z})$ be such that ${}^tFF=\ell\,\mathrm{Id}$.

Let $i\in K_1(\mathscr{L})$ and $(j_1,\dots,j_r)\in K_1(\mathscr{M})^r$ be the unique preimage of $(i,0,\dots,0)$ by F. Let y be a geometric point of B and let $Y={}^tF(y,0,\dots,0)\in B^r$. Then (up to a constant λ)

$$\vartheta_i^{\mathscr{L}}(\widetilde{f}(y))\cdot\dots\cdot\vartheta_0^{\mathscr{L}}(0)=\lambda\sum_{\substack{(t_1,\dots,t_r)\in K'^r\\ F(t_1,\dots,t_r)=(0,\dots,0)}}\vartheta_{j_1}^{\mathscr{M}}(Y_1+t_1)\cdot\dots\cdot\vartheta_{j_r}^{\mathscr{M}}(Y_r+t_r). \tag{1}$$

Proof : From the hypothesis ℓ prime to $2n$, Theorem 2.2 show that every compatible symmetric theta structure on $G(\mathscr{M}^\ell)$ induce the same totally symmetric line bundle $\mathscr{L}$ on A and the induced symmetric theta structure on $G(\mathscr{L})$ depends only on the choice of the symmetric theta structure on $G(\mathscr{M})$.

Now we just need to apply the diagram from above. In this diagram we apply Proposition 3.1 with $X={}^tF(x,0,\dots,0)$ where $x\in A$ is such that $f(x)=y$.

Now since ℓ is prime to n, an element $h\in K_1(\mathscr{L}^\ell)$ is of the form $h=j+T$ where $j\in K_1(\mathscr{L})$ and $T\in K_1(\mathscr{L}^\ell)[\ell]$. But by Proposition 3.2, $\vartheta_h^{\mathscr{L}^\ell}(X_i)=\vartheta_{f(j)}^{\mathscr{M}}(Y_i+f(T))$ (think about our $g=1$, $n=2$, $\ell=3$ example. Looking at the equation in Proposition 3.1 we get Equation 1.

Note that in Equation 1, the coordinates of the right hand term are not the projective coordinates of the points (it would not make sense in a sum) but of suitably normalized affine lifts. More details are given in [CR11] where we explain how to use differential additions to normalize the affine lifts.

In total, the complexity cost is given by normalizing affine lifts of the geometric points of K $O(\ell^g)$ and the changing level formula costing $O(\ell^{gr/2})$. (For an improvement, in [Rob12] we explain with David Lubicz how to adapt the formula to only need the equations of the kernel K). ■

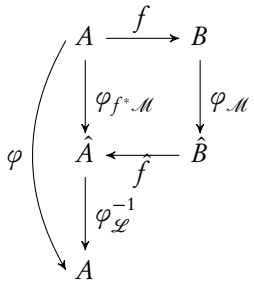
Example 3.5 :
If $\ell=a^2+b^2$, we can take $F=\begin{pmatrix}a & b\\ -b & a\end{pmatrix}$, so that Equation 1 become

$$\vartheta_i^{\mathscr{L}}(f(y))\cdot\vartheta_0^{\mathscr{L}}(0)=\lambda\sum_{t\in K}\vartheta_{j_1}^{\mathscr{M}}(ay+at)\cdot\vartheta_{j_2}^{\mathscr{M}}(by+bt). \tag{2}$$

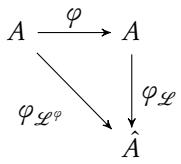
4 Computing isogenies with cyclic kernel

Let $f:A\to B$ be an isogeny with cyclic kernel, and assume that we have principal polarization $\mathscr{L}_0$ and $\mathscr{M}_0$ on A and B. Let $\mathscr{L}=\mathscr{L}_0^n$ and $\mathscr{M}=\mathscr{M}_0^n$.

Then there exist φ such that the following diagram commutes:



By construction, φ commutes with the Rosatti involution, so it is a (totally positive) totally real element of $\mathrm{End}^0(A)$. We note $\mathscr{L}^\varphi = f^*\mathscr{M}$ so that we have the following diagram



Since the commutator pairing $e_{\mathscr{L}^\varphi}$ is non degenerate (or since $\mathrm{Ker}\,\hat{f}$ is the Cartier dual of $K = \mathrm{Ker}\, f$), we see that $\mathrm{Ker}\,\varphi \subset A[\ell]$ is non isotropic for the Weil pairing. However, $K = \mathrm{Ker}\, f$ is maximally isotropic for $e_{\mathscr{L}^\varphi}$. So in Section 3 we explained how to compute an isogeny from a maximal isotropic kernel K (implicitly for Weil pairing $e_{\mathscr{L}^\ell}$), this suggest that we will be able to compute the isogeny with kernel K maximally isotropic for $e_{\mathscr{L}^\varphi}$ by replacing $[\ell]$ with φ everywhere.

Of course at one point we will need to explain how to construct $\mathscr{L}^\varphi$ without using the isogeny f, because we want to compute f from $\mathscr{L}^\varphi$.

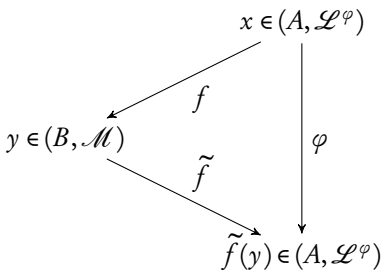
First, the analog of Proposition 3.2 is immediate:

Proposition 4.1 :
Assume that the symmetric theta structure on $\mathscr{L}^\varphi$ is such that $K \subset K_2(\mathscr{L}^\varphi)$ (this is always possible). Then the symmetric theta structure on $\mathscr{L}^\varphi$ induces a symmetric theta structure on $\mathscr{M}$ by Theorem 2.2 (this may require to replace $\mathscr{M}$ by an equivalent line bundle).

Let $f : A \to B$ be the isogeny of kernel K, and x a geometric point in A. Fix $i \in K_1(\mathscr{M})$, and let $j \in K_1(\mathscr{L}^\varphi)$ be the unique preimage of i by f that is in $K_1(\mathscr{L}^\varphi)$. We have (up to a constant λ)

$$\vartheta_i^{\mathscr{M}}(f(x)) = \lambda \vartheta_j^{\mathscr{L}^\varphi}(x)$$

Moreover, if we introduce the φ-contragredient isogeny $\widetilde{f}$ has the isogeny $\widetilde{f} : B \to A$ such that $\widetilde{f} \circ f = \varphi$, we have the following diagram



The exact same techniques as in Section 3 allow to find from $y \in B$ a preimage x, and such compute $\widetilde{f}(y)$ in coordinates from $\mathscr{L}^\varphi$. Now we just need to apply a change level formula using an equivalent of Proposition 4.

First we need to find an equivalent of the matrix F. To simplify we now assume that the division algebra $\mathrm{End}^0(A)$ is a field K, and we let K_0 be the associated totally real field. Furthermore, we assume that $O_0 = K_0 \bigcap \mathrm{End}(A)$ is the maximal order O_{K_0} of K_0 (A has maximum real multiplication).

Lemma 4.2 :
Let $\varphi \in O_{K_0}$ be a totally positive element. Then there exist $F \in \mathrm{Mat}_r(O_{K_0})$ such that ${}^tFF = \varphi\,\mathrm{Id}$.

Proof : It is well known that such a φ is a sum of m squares in O_{K_0}. We may assume that $m = 2^d$ is a power of 2. Now using the theory of Clifford's algebra for the quadratic form $Q(x_1,\dots,x_t) = -x_1^2 - x_2^2 - \cdots - x_t^2$ with $t \geqslant d$ sufficiently large, we obtain the matrix F with $r = 2^t$.

[Update 2013-04-23: as remarked by Dimitar Jetchev, a paper of Siegel show that except in $Q(\sqrt{5})$ for some elements of O_{K_0} a sum of squares can only be found using non integral elements. If we have such an element α/m, to compute its action on the ℓ-torsion, we need to compute the action of α on the ℓm-torsion, so we would like m to be as small as possible. Intuitively, for a larger r we can get a smaller m, but a large r also increase the complexity.] ■

Remark 4.3 :

- φ is a sum of two squares iff it is the norm of an element of $K_0(i)$. This is purely a local question, so it should be pretty easy to test in practice.
- In general, $\mathbb{Q}(\sqrt{5})$ is the only real quadratic field whose every integral element is a sum of 4 integral squares [TODO: check if this is correct] . So we way need to take $d > 2$.
- Also, the generic formula converting a sum of 2^d squares into a matrix of length 2^d involves denominator. That's why in the proof of the lemma we need to assume that t may be larger than d (the exact formula is given by the size of the representations of the associated Clifford's algebra).
- Still, the following will make clear that we only need to work locally on $\mathbb{Z}[\frac{1}{2\ell n}]$ so we can look for F in $\mathrm{Mat}_r(O_{K_0} \otimes \mathbb{Z}[\frac{1}{2\ell n}]$.
- All in all, I lack a clear bound on how big r could be at worse. Note that the size of r directly influence the cost of the changing level formulas (see Proposition 3.4).
- To look for smaller r, Christophe Ritzenthaler suggested looking at matrix F such that (for instance) ${}^tFF = \mathrm{diag}(\varphi, 1, \dots, 1)$. ♢

Now assume the matrix F is fixed, we have

Proposition 4.4 :
Let $(i_1,\dots,i_r) \in K_1(\mathscr{L})^r$. Let $x = (x_1,\dots,x_r)$ be a geometric point of A^r and let $y = Fx$. Then (up to a constant λ)

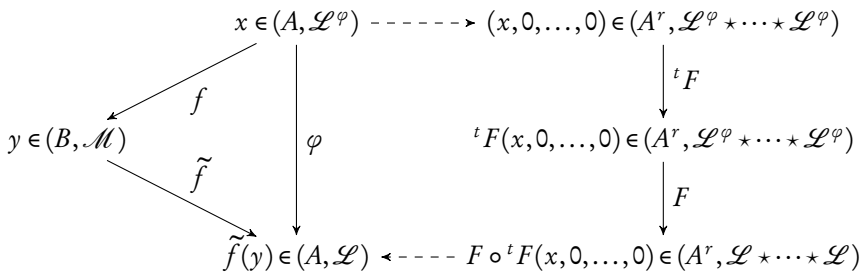
$$\vartheta^{\mathscr{L}}_{i_1}(y_1)\cdot\dots\cdot\vartheta^{\mathscr{L}}_{i_r}(y_r) = \lambda \sum_{\substack{(j_1,\dots,j_r)\in K_1(\mathscr{L}^\varphi)\\ F(j_1,\dots,j_r)=(i_1,\dots,i_r)}} \vartheta^{\mathscr{L}^\varphi}_{j_1}(x_1)\cdot\dots\cdot\vartheta^{\mathscr{L}^\varphi}_{j_r}(x_r).$$

Proof : It is a bit easier to look at a proof over $\mathbb{C}$. The action of F on the polarization H is given by ${}^t\overline{F}F = \varphi\,\mathrm{Id}$ (because the elements of F are real), so we have $F^*\mathscr{L} \star \mathscr{L} \cdots = \mathscr{L}^\varphi \star \mathscr{L}^\varphi \cdots$.

Note that this give the construction of $\mathscr{L}^\varphi$ we were looking for. Now the real elements of K_0 acts on $\mathbb{C}^g$ by symmetric matrixes, so they are codiagonalizable in respect to the principal polarization H_0.

In particular, the isogeny induced by F on A^r respect the symplectic decomposition given on A, so we can apply the isogeny theorem. ■

Now we just have to combine everything in the following diagram

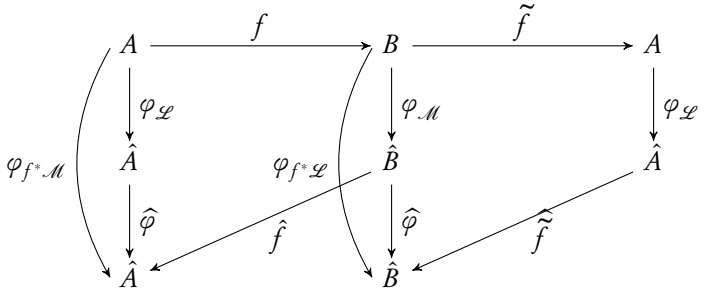


Proposition 4.5 :
Let $(B, \mathscr{M}_0)$ be a ppav with a symmetric theta structure on $G(\mathscr{M})$ where $\mathscr{M} = \mathscr{M}_0^n$ is of level n even. Let $K' \subset B[\ell]$ be a maximal isotropic subgroup for $\mathscr{M}^\varphi$ and $\widetilde{f} : B \to A = B/K'$ be the associated isogeny. Assume that ℓ is prime to $2n$; then the theta structure on $G(\mathscr{M})$ induces a unique polarization $\mathscr{L}$ of level n on A and a unique compatible symmetric theta structure on $G(\mathscr{L})$. Let $F \in \mathrm{Mat}_r(O_{K_0})$ be such that ${}^tFF = \varphi\,\mathrm{Id}$.

Let $i \in K_1(\mathscr{L})$ and $(j_1, \dots, j_r) \in K_1(\mathscr{M})^r$ be the unique preimage of $(i,0,\dots,0)$ by F. Let y be a geometric point of B and let $Y = {}^tF(y,0,\dots,0) \in B^r$. Then (up to a constant λ that may depend on y this time)

$$\vartheta_i^{\mathscr{L}}(\widetilde{f}(y)) \cdot \dots \cdot \vartheta_0^{\mathscr{L}}(0) = \lambda \sum_{\substack{(t_1,\dots,t_r) \in K'^r \\ F(t_1,\dots,t_r) = (0,\dots,0)}} \vartheta_{j_1}^{\mathscr{M}}(Y_1 + t_1) \cdot \dots \cdot \vartheta_{j_r}^{\mathscr{M}}(Y_r + t_r). \tag{3}$$

Note that the condition of having maximal real multiplication is too strong, we only need to have a matrix F corresponding to φ. In particular, we don't really need to have maximal real multiplication, nor even that A and B have the same real multiplication. Of course, we do need to have φ in $\mathrm{End}(A)$ and $\mathrm{End}(B)$, where we abuse the same notation to note $\varphi = \widetilde{f} \circ f \in \mathrm{End}(A)$ and $f \circ \widetilde{f} \in \mathrm{End}(B)$. Perhaps the following diagram is clearer:



4.1 Computing the isogeny in practice

Of course in Proposition 4.5 we have hidden all the difficulties in the computation of

$$\vartheta_{j_1}^{\mathscr{M}}(Y_1 + t_1) \cdot \dots \cdot \vartheta_{j_r}^{\mathscr{M}}(Y_r + t_r),$$

where we need to have a way to compute the action of the elements of O_{K_0} giving F in a "compatible affine manner".

The easy case is if we only need the isogenous theta null point. In which case $y = 0$ and $Y = (0,\dots,0)$ so that we only need to evaluate on points of K' but we have already seen how to normalize the affine lifts [CR11]. But to compute the image of a point y we need to work harder.

We give an example on how to do that with $O_{K_0} = \mathbb{Q}(\sqrt{d})$ (d prime to ℓ) and $\varphi = a^2 + b^2$ (the generalization to a sum of more squares is immediate) so that we can take $F = \begin{pmatrix} a & b \\ -b & a \end{pmatrix}$ as in Example 3.5 We need to evaluate

$$\sum_{t \in K} \vartheta_{j_1}^{\mathscr{M}}(ay + at) \cdot \vartheta_{j_2}^{\mathscr{M}}(by + bt). \tag{4}$$

We want to compute affine coordinates of $ay+at$ and $by+bt$, where the eventual projective factor depends only on y, not on t. Let $a=\alpha+\beta\sqrt{d}$ and let's see what we can compute.

Since we have normalized all the points of K', we know αt, $\beta\sqrt{d}t$ and $\alpha+\beta\sqrt{d}$ already. We also know the "affine coordinates" of αy and $\alpha(y+t)$, this only use differential additions.

We also can compute $\sqrt{d}y$ since $\sqrt{d}$ correspond to a (d,d)-isogeny (a normal one with maximal isotropic kernel for the Weil pairing). The important point here is that Proposition 3.4 gives us the isogeny for **affine theta coordinates** (since λ is a constant). From $\sqrt{d}y$ we get $\beta\sqrt{d}y$ using differential additions. Likewise we can compute $\beta\sqrt{d}(y+t)$. If $\alpha t=\beta\sqrt{d}t'$, then $\beta\sqrt{d}y+\alpha t$ is simply $\beta\sqrt{d}(y+t')$ so we can also compute it.

Finally we can compute $\alpha+\beta\sqrt{d}y$ but only in a projective way, so we have take an arbitrary affine lift. The important point here is that we can fix it once and for all, it does not depend on t.

In the sum of four terms $\alpha y+\beta\sqrt{d}y+\alpha t+\beta\sqrt{d}t$, we have seen how to compute each of the two by two subsum. Now this is what we call a MultiWayAddition, and we claim that by using Riemann relations, this is enough to compute the whole sum. Indeed, it is easy to see that a MultiWayAddition reduces to several ThreeWayAdditions (compute $x+y+z$ from x, y, z, $x+y$, $x+z$, $y+z$) and we showed how to do that in [Rob10; LR13] (generically in level 2, for any geometric point in level $n>2$.)

Remark 4.6 :

- Finding the matrix F requires that we know what the full real endomorphism order look like, which may be expensive. Over a finite field, it should be possible by Tate's theorem to work on the ℓ-Tate module to find the action of F on the ℓ-torsion, which is what we need if we only want the isogenous theta null point (we also need the action on the 2-torsion, so we'll need to glue things).
- It would be interesting to have a purely analytic version of Proposition 4.5. Note that the analytic version of Koizumi [Koi76] is a bit stronger as stated than the algebraic version of Kempf [Kem89] (for instance to recover the usual Riemman's relation, one need to apply Kempf's version twice). ♢

References

[BL04] C. Birkenhake and H. Lange. *Complex abelian varieties*. Second. Vol. 302. Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]. Berlin: Springer-Verlag, 2004, pp. xii+635. ISBN: 3-540-20488-1 (cit. on pp. 3, 6).

[BCR10a] G. Bisson, R. Cosset, and D. Robert. "AVIsogenies (Abelian Varieties and Isogenies)". Packet magma dédié au calcul explicite d'isogénies entre variétés abéliennes. 2010. URL: http://avisogenies.gforge.inria.fr. Licence libre (LGPLv2+), enregistré à l'APP (référence IDDN.FR.001.440011.000.R.P.2010.000.10000) (cit. on p. 1).

[BCR10b] G. Bisson, R. Cosset, and D. Robert. "On the Practical Computation of Isogenies of Jacobian Surfaces". Article explaining the computations done in the AVIsogenies package, found in the source code. 2010. URL: http://avisogenies.gforge.inria.fr (cit. on p. 1).

[Cos11] R. Cosset. "Application des fonctions thêta à la cryptographie sur courbes hyperelliptiques". PhD thesis. 2011 (cit. on pp. 1, 7).

[CR11] R. Cosset and D. Robert. "An algorithm for computing (ℓ,ℓ)-isogenies in polynomial time on Jacobians of hyperelliptic curves of genus 2". Mar. 2011. URL: http://www.normalesup.org/~robert/pro/publications/articles/niveau.pdf. HAL: hal-00578991, eprint: 2011/143 (cit. on pp. 1, 2, 4, 8, 11).

[FLR11] J.-C. Faugère, D. Lubicz, and D. Robert. "Computing modular correspondences for abelian varieties". In: *Journal of Algebra* 343.1 (Oct. 2011), pp. 248–277. DOI: 10.1016/j.jalgebra.2011.06.031. arXiv: 0910.4668 [cs.SC]. URL: http://www.normalesup.org/~robert/pro/publications/articles/modular.pdf. HAL: hal-00426338 (cit. on pp. 1, 6).

[Igu72] J.-i. Igusa. *Theta functions*. Die Grundlehren der mathematischen Wissenschaften, Band 194. New York: Springer-Verlag, 1972, pp. x+232 (cit. on p. 3).

[Kem89] G. Kempf. "Linear systems on abelian varieties". In: *American Journal of Mathematics* 111.1 (1989), pp. 65–94 (cit. on pp. 1, 4, 12).

[Koi76] S. Koizumi. "Theta relations and projective normality of abelian varieties". In: *American Journal of Mathematics* (1976), pp. 865–889 (cit. on pp. 1, 3, 12).

[LR12a] K. Lauter and D. Robert. "Improved CRT Algorithm for class polynomials in genus 2". In: *ANTS* (2012). Accepted for publication at the Tenth Algorithmic Number Theory Symposium ANTS-X. University of California, San Diego, July 9 – 13, 2012 http://math.ucsd.edu/~kedlaya/ants10/. URL: http://www.normalesup.org/~robert/pro/publications/articles/classCRT.pdf. Slides http://www.normalesup.org/~robert/publications/slides/2012-07-ANTS-SanDiego.pdf, eprint: 2012/443, HAL: hal-00734450 (cit. on p. 2).

[LR12b] D. Lubicz and D. Robert. "Computing isogenies between abelian varieties". In: *Compositio Mathematica* 148.05 (Sept. 2012), pp. 1483–1515. DOI: 10.1112/S0010437X12000243. arXiv: 1001.2016 [math.AG]. URL: http://www.normalesup.org/~robert/pro/publications/articles/isogenies.pdf. HAL: hal-00446062 (cit. on pp. 1, 7).

[LR13] D. Lubicz and D. Robert. "A generalisation of Miller's algorithm and applications to pairing computations on abelian varieties". Mar. 2013. URL: http://www.normalesup.org/~robert/pro/publications/articles/optimal.pdf. HAL: hal-00806923, eprint: 2013/192 (cit. on p. 12).

[Mum66] D. Mumford. "On the equations defining abelian varieties. I". In: *Invent. Math.* 1 (1966), pp. 287–354 (cit. on pp. 3, 4, 6).

[Mum67a] D. Mumford. "On the equations defining abelian varieties. II". In: *Invent. Math.* 3 (1967), pp. 75–135 (cit. on pp. 3, 6).

[Mum67b] D. Mumford. "On the equations defining abelian varieties. III". In: *Invent. Math.* 3 (1967), pp. 215–244 (cit. on p. 3).

[Mum70] D. Mumford. *Abelian varieties*. Tata Institute of Fundamental Research Studies in Mathematics, No. 5. Published for the Tata Institute of Fundamental Research, Bombay, 1970, pp. viii+242 (cit. on p. 3).

[Mum83] D. Mumford. *Tata lectures on theta I*. Vol. 28. Progress in Mathematics. With the assistance of C. Musili, M. Nori, E. Previato and M. Stillman. Boston, MA: Birkhäuser Boston Inc., 1983, pp. xiii+235. ISBN: 3-7643-3109-7 (cit. on p. 3).

[Mum84] D. Mumford. *Tata lectures on theta II*. Vol. 43. Progress in Mathematics. Jacobian theta functions and differential equations, With the collaboration of C. Musili, M. Nori, E. Previato, M. Stillman and H. Umemura. Boston, MA: Birkhäuser Boston Inc., 1984, pp. xiv+272. ISBN: 0-8176-3110-0 (cit. on p. 3).

[Mum91] D. Mumford. *Tata lectures on theta III*. Vol. 97. Progress in Mathematics. With the collaboration of Madhav Nori and Peter Norman. Boston, MA: Birkhäuser Boston Inc., 1991, pp. viii+202. ISBN: 0-8176-3440-1 (cit. on p. 3).

[Rob10] D. Robert. "Fonctions thêta et applications à la cryptographie". PhD thesis. Université Henri-Poincarré, Nancy 1, France, July 2010. URL: http://www.normalesup.org/~robert/pro/publications/academic/phd.pdf. Slides http://www.normalesup.org/~robert/pro/publications/slides/2010-07-phd.pdf, TEL: tel-00528942. (Cit. on pp. 1, 4, 7, 12).

[Rob12] D. Robert. "Computing rational isogenies from the equations of the kernel". ANR Peace Meeting, Paris. Nov. 2012. URL: http://www.normalesup.org/~robert/pro/publications/slides/2012-11-Peace.pdf (cit. on pp. 1, 8).

[Rob13] D. Robert. "Computing cyclic isogenies using real multiplication". ANR Peace Meeting, Paris. Notes available on http://www.normalesup.org/~robert/pro/publications/notes/2013-04-cyclic-isogenies.pdf. Apr. 2013 (cit. on p. 1).

[Shi98] G. Shimura. *Abelian varieties with complex multiplication and modular functions*. Vol. 46. Princeton University Press, 1998 (cit. on p. 2).

[Vél71] J. Vélu. "Isogénies entre courbes elliptiques". In: *Compte Rendu Académie Sciences Paris Série A-B* 273 (1971), A238–A241 (cit. on p. 1).

[Wam99] P. Wamelen. "Equations for the Jacobian of a hyperelliptic curve". In: *AMS* 350.8 (Aug. 1999), pp. 3083–3106 (cit. on p. 1).

2 ON SYMMETRIC THETA STRUCTURES (2013-04)

These notes are available at http://www.normalesup.org/~robert/pro/publications/notes/2013-04-theta-sym.pdf.

The purpose of these short notes is to highlight the role of symmetry and symmetric theta structures in Mumford's isogeny theorem.

On symmetric theta structures

Damien Robert

April 26, 2013

1 Symmetric theta structures and the isogeny theorem

Let A be an abelian variety of dimension g defined over an algebraically closed field $\overline{k}$. Let $\mathscr{L}_0$ be a symmetric ample line bundle of degree one on A, $\mathscr{L}_0$ defines a principal polarization: $A \to \hat{A}$. If n is even $\mathscr{L} = \mathscr{L}_0^n$ is then totally symmetric, and the kernel $K(\mathscr{L})$ of the polarization associated to $\mathscr{L}$ is $A[n]$.

From now on, we assume that n is prime to the characteristic of k, so that $\mathscr{L}$ defines a separable polarisation. Since $\mathscr{L}$ is totally symmetric, there exist a symmetric theta structure on the theta group $G(\mathscr{L})$. Fixing such a structure fix a unique projective basis of theta functions [Mum66] that we call theta functions of level n. Note: the theta structure induces an isomorphism between the symplectic spaces $Z(\overline{n}) \times \hat{Z}(\overline{n})$ and $K(\mathscr{L}) = A[n]$ where $Z(\overline{n}) = (\mathbb{Z}/n\mathbb{Z})^g$ and $\hat{Z}(\overline{n})$ is the Cartier dual of $Z(\overline{n})$. We note $K(\mathscr{L}) = K_1(\mathscr{L}) \oplus K_2(\mathscr{L})$ where $K_1(\mathscr{L})$ corresponds to $Z(\overline{n})$ and $K_2(\mathscr{L})$ to $\hat{Z}(\overline{n})$. Usually the canonical basis of the theta functions of level n are indexed by $i \in Z(\overline{n})$, but in these notes we will index them by $i \in K_1(\mathscr{L})$ which permit us to not track explicitly the isomorphism between $Z(\overline{n})$ and $K_1(\mathscr{L})$.

If $n > 2$ then the theta functions of level n give a projective embedding of A into $\mathbb{P}^{n^g-1}_{\overline{k}}$, while if $n = 2$ we only get an embedding of the Kummer variety $A/\pm 1$ (the $n = 2$ case assume that A is absolutely simple, see [BL04]). Under a generic condition (the even theta null coordinates are non zero), this embedding of the Kummer variety is actually projectively normal (see [Koi76]).

Theorem 1.1 :
The symmetric theta structure on $G(\mathscr{L})$ is uniquely determined by a choice of symplectic basis $(e_1, \dots e_g, e'_1, \dots e'_g)$ on $A[n]$ and a choice of symplectic basis $(f_1, \dots f_g, f'_1, \dots f'_g)$ on $A[2n]$ such that $e_i = 2f_i, e'_i = 2f'_i$. (Here symplectic mean for the commutator pairing $e_{\mathscr{L}}$ and $e_{\mathscr{L}^2}$ respectively).

Moreover, changing these symplectic basis do not change the resulting symmetric theta structure if and only if

- *The symplectic basis of $A[n]$ is left invariant;*
- *The f_i are replaced by points $f_i + t_i$ with $t_i \in A[2]$ such that $e_{\mathscr{L}}(e_i, t_i) = 1$.*

In particular, fixing a symplectic basis of $A[n]$ and a symplectic decomposition $A[2n] = A_1[2n] \oplus A_2[2n]$ of the $2n$-torsion into a sum of maximal isotropic subspaces is enough (and even stronger) to fix the symmetric theta structure.

Proof : This is implicit in [Mum66, Section 3]. A symmetric theta structure comes from an isomorphism between the Heisenberg group and the theta group that commutes with the action of $[-1]$. It induces an isomorphism between the symplectic spaces $Z(\overline{n}) \times \hat{Z}(\overline{n})$ and $K(\mathscr{L}) = A[n]$ and hence fix a symplectic basis of the n-torsion.

Conversely, having fixed a symplectic basis of the n-torsion, since $\mathscr{L}$ is totally symmetric, there is always a symmetric theta structure respecting this symplectic basis. Such a choice of a symmetric theta structure can be seen as a choice of a symmetric element above each of the element of the basis $(e_1, \dots e'_g)$; since there is only two symmetric elements $\pm g_i$ above each e_i a symmetric theta structure above the symplectic basis can be seen as a choice of sign for each element of the basis.

If $g_i \in G(\mathscr{L}^2)$ is a symmetric element of the theta group above a point f_i such that $e_i = 2f_i$, then $(g_i)^2$ determines a symmetric element of the theta group above e_i that uniquely depends on the choice of f_i (since the other symmetric element above f_i is $-g_i$ which gives rise to $(-g_i)^2 = (g_i)^2$ above e_i. Via the transfer map δ_2 from [Mum66], we see how the choices of the f_i above the e_i are enough to determine the symmetric theta structure on $G(\mathscr{L})$.

It is a straightforward verification to see that replacing f_i by $f_i + t_i$ where t_i is a point of 2-torsion involve replacing $(g_i)^2$ by $e_{\mathscr{L}^2}(f_i, t_i)(g_i)^2$ which concludes the proof.

(One could also replace the application δ_2 by the isogeny $[2]$ which would involve working in $G(\mathscr{L}^4)$, as in [Kem89].) ■

Corollary 1.2 :
Let $(A, \mathscr{L}_0)/\mathbb{F}_q$ be a ppav over the finite field $\mathbb{F}_q$. Assume that $\mu_n(\overline{\mathbb{F}}_q) \subset \mathbb{F}_q$ ($n = 2n_0$ even). Then there exist a rational symmetric theta structure on $\mathscr{L} = \mathscr{L}_0^n$ iff there exist a rational symplectic basis $(e_1, \dots e_g, e'_1, \dots, e'g)$ such that $e_{T,2}(n_0 e_i, e_i) = 1$; where $e_{T,2}$ denotes the 2-Tate pairing. (In other words, e_i form a symplectic basis consisting of elements whose self n-Tate pairing is not a primitive n-th root of unity).

Proof : This is clear from Theorem 1.1 and the definition of the Tate pairing as $e_{T,2}(n_0 e_i, e_i) = e_{W,2}(n_0 e_i, \pi(f_i) - f_i)$ where $2f_i = e_i$ and π is the Frobenius of $\mathbb{F}_q$. ■

Remark 1.3 :
In the case that $\mathbb{F}_q$ does not contain the n-th root of unity, a rational theta structure of level n induces an equivariant (for the Galois action) isomorphism between $A[n]$ and $Z(\overline{n}) \times \hat{Z}(\overline{n})$. In particular, this does not impose that all geometric points of $A[n]$ are rational.

Proposition 1.4 :
Let $\mathscr{L}$ be a symmetric line bundle on A, defining a polarization of type $\delta = (\delta_1, \dots, \delta_g)$. Then there exists a symmetric theta structure on $G(\mathscr{L})$ if and only if for every $x \in A[2] \bigcap K(\mathscr{L})$, we have $e_*(x) = 1$.

In this case we call $\mathscr{L}$ totally symetrisable (because a totally symmetric line bundle satisfy the condition), and the obvious generalisation of Theorem 1.1 to this case also holds.

Proof : [Kem89; Mum66]. ■

The idea is that (for instance in dimension 2), $\mathscr{L}_0^\ell$ is of type (ℓ, ℓ) and allows to compute isogenies with maximal isotropic kernels, but for a cyclic isogeny we need a polarisation of type $(1, \ell)$ (like the type of $\mathscr{L}_0^\rho$ from Section **??**).

Theorem 1.5 :
Let $f : (A, \mathscr{L}) \to (B, \mathscr{M})$ be an isogeny between pav. Then $K = \operatorname{Ker} f$ is isotropic in $K(\mathscr{L})$ for the commutator pairing $e_{\mathscr{L}}$, and $K(\mathscr{M}) \simeq K^\perp / K$.

Assume that we have a symmetric theta structure on $G(\mathscr{L})$ coming from a symplectic basis (f_i, f'_i) on $K(\mathscr{L}^2)$. Assume that K is compatible with the induced symplectic decomposition $K(\mathscr{L}) = K_1(\mathscr{L}) \oplus K_2(\mathscr{L})$ into maximal isotropic subspaces in the sense that $K = K_1 \oplus K_2$ where $K_i = K_i(\mathscr{L}) \bigcap K$. In this case $K(\mathscr{M}) \simeq K^{2,\perp}/K_1 \oplus K^{1,\perp}/K_2$ where $K^{2,\perp} = K_2^\perp \bigcap K_1(\mathscr{L})$ and $K^{1,\perp} = K_1^\perp \bigcap K_2(\mathscr{L})$

Let $\widetilde{K}$ be the level subgroup above K induced by this theta structure; the corresponding descent data give a line bundle $\mathscr{M}'$ algebraically equivalent to $\mathscr{M}$. Moreover $\mathscr{M}'$ is totally symetrisable, and we can define a symmetric theta structure on $\mathscr{M}'$ as follow: from the symplectic basis of $K(\mathscr{L}^2)$ one derives a "canonical" basis $(g_1, \dots, g'_g)$ of $[2]^{-1}K^\perp$. Pushing this basis via the isogeny f gives a symplectic basis on $K(\mathscr{M}'^2)$, which determines the symmetric theta structure on $\mathscr{M}'$. It is easy to see that by construction, it is compatible with the theta structure on $\mathscr{L}$.

We can then apply the isogeny theorem: there exist λ such that for all $i \in K_1(\mathscr{M}')$

$$\vartheta_i^{\mathscr{M}'} = \lambda \sum_{j \in K_1(\mathscr{L}) | f(j) = i} \vartheta_j^{\mathscr{L}}.$$

Proof : [Kem89; Mum66; Rob10]. ■

Corollary 1.6 :

- *If $\mathscr{M}$ is of type δ' with $2 \mid \delta'$ (meaning that $A[2] \bigcap K(\mathscr{L}) \subset K^{\perp}$), then $\mathscr{M}'$ is the unique totally symmetric line bundle in the equivalence class of $\mathscr{M}$.*
- *If $A[2] \bigcap K(\mathscr{L}) \subset K$, then every symmetric theta structure on $G(\mathscr{L})$ induces the same symmetric theta structure on $G(\mathscr{M}')$.*

Proof : See [Kem89; Rob10]. ■

3 ISOGENIES BETWEEN ABELIAN VARIETIES (2014-06)

These notes are available at http://www.normalesup.org/~robert/pro/publications/notes/2014-06-Rennes-Moduli.pdf.

They were meant as a summary of the results of [FLR11; LR12; CR15; DJRV22]. These don't include the last improvements [KPR20; LR22; DMPR23a], and the treatment in [Rob21a] is more complete. The corresponding implementation is AVIsogenies [BCR10].

Isogenies between abelian varieties

DAMIEN ROBERT

Notes of a talk given for the Conference "Effective moduli spaces and applications to cryptography" — Rennes

Abstract. In this talk we give a brief panorama of the effective computation of isogenies between principally polarized abelian varieties and of modular equations.

Given a principally polarized abelian variety A, we want to compute the following:

- Given a kernel K, compute the isogeny $A \to B = A/K$;
- Given a degree ℓ, compute all isogenous abelian varieties B (where the isogeny is of degree ℓ^g);
- Given two abelian varieties A and B, test if they are isogenous (of a given degree). If so find the kernel K of the isogeny $A \to B$.

Note: We will restrict to perfect fields, separable isogenies and principally polarised abelian varieties. In particular we will deal with isotropic kernels.

Theorem 0.1. *Suppose that A/k, B/k are absolutely simple over a perfect field k. Suppose that $\mathrm{Hom}_k(A,B) \neq \{0\}$, $\mathrm{End}_k(A) = \mathrm{End}_{\overline{k}}(A)$. Then $\mathrm{Hom}_k(A,B) = \mathrm{Hom}_{\overline{k}}(A,B)$.*

1. Elliptic curves

1.1. Isogenies from the kernel.

Theorem 1.1 ([Vél71])**.** *Let $E : y^2 = f_1(x)$ be an elliptic curve and $K \subset E(k)$ a finite subgroup. Then E/K is given by $Y^2 = f_2(X)$ where*

$$X(P) = x(P) + \sum_{Q \in K \setminus \{0_E\}} (x(P+Q) - x(Q))$$
$$Y(P) = y(P) + \sum_{Q \in K \setminus \{0_E\}} (y(P+Q) - y(Q)).$$

If $f_1(x) = x^3 + ax + b$ then $f_2(x) = x^3 + (a - 5t)x + b - 7w$ where

$$t = \sum_{Q \in K \setminus \{0_E\}} f'(Q), \quad u = 2 \sum_{Q \in K \setminus \{0_E\}} f(Q), \quad w = \sum_{Q \in K \setminus \{0_E\}} x(Q) f'(Q).$$

Proof. Uses the fact that x and y are characterised in $k(E)$ by

$$v_{0_E}(x) = -2 \qquad v_P(x) \geq 0 \quad \text{if } P \neq 0_E$$
$$v_{0_E}(y) = -3 \qquad v_P(y) \geq 0 \quad \text{if } P \neq 0_E$$
$$y^2/x^3(0_E) = 1$$

□

Theorem 1.2 ([Koh96])**.** *Let $h(x) = \prod_{Q \in K \setminus \{0_E\}} (x - x(Q))$ defining the subgroup K of the elliptic curve $E_1 : y^2 = f_1(x)$; then the isogeny $f : E_1 \to E_2$ is defined by*

$$f(x,y) = \left(\frac{g(x)}{h(x)}, y \left(\frac{g(x)}{h(x)} \right)' \right), \text{ with } \frac{g(x)}{h(x)} = \#K.x - \sigma - f'(x)\frac{h'(x)}{h(x)} - 2f(x)\left(\frac{h'(x)}{h(x)} \right)',$$

Date: June 10, 2014.

where σ is the first power sum of h (the sum of the x-coordinates of the points in the kernel). When $\#K$ is odd, $h(x)$ is a square, so we can replace it by its square root. The complexity of computing the isogeny is then $O(M(\#K))$ operations in k.

Proof. Let $w_{E_1} = dx/2y$ be the canonical differential. Then $f^*w_{E_2} = cw_{E_2}$, with c in k. Up to a normalisation we can assume that $c = 1$, so $f(x,y) = \left(\frac{g(x)}{h(x)}, y\left(\frac{g(x)}{h(x)}\right)'\right)$. Plugging the formulas from theorem 1.1 yields the result. □

To compute all rational isogenous elliptic curves starting from E_1 with an isogeny of degree ℓ, we can compute all rational cyclic subgroups of $E[\ell]$ and apply Vélu's formulas. These subgroups can be obtained as factors of the ℓ-division polynomial $\prod_{Q\in E[\ell]\setminus\{0_E\}}(x - x(Q))$. This division polynomial has degree $(\ell^2-1)/2$ (if ℓ odd), and factorizing it will cost $O(\ell^{3.63})$ (over a finite field).

1.2. **Modular polynomials.**

Definition 1.3. Modular polynomials The modular polynomial $\varphi_\ell(x_1,x_2) \in \mathbb{Z}[x_1,x_2]$ is a bivariate polynomial such that $\varphi_\ell(x_1,x_2) = 0 \Leftrightarrow x = j(E_1)$ and $y = j(E_2)$ with E_1 and E_2 ℓ-isogenous.

One can also see the modular polynomial as the polynomial describing the modular curve $X_0(\ell)$ inside $X(1)\times X(1) \simeq \mathbb{P}^2$ [Koh03].

Proposition 1.4. *φ_ℓ is a symmetric polynomial of degree $\ell+1$. The height of the coefficients of φ_ℓ grows as $O(\ell\log\ell)$.*

The roots of $\varphi_\ell(j(E_1),.)$ are exactly the elliptic curves ℓ-isogenous to E_1. There are $\ell+1 = \#\mathbb{P}^1(\mathbb{F}_\ell)$ such roots if ℓ is prime.

Theorem 1.5 (Rational roots of modular polynomials)**.** *Let $E_1/\mathbb{F}_q$ be an ordinary elliptic curve, ℓ be a prime and j_2 be a root of $\varphi_\ell(j_{E_1},\cdot)$ over $\mathbb{F}_{q^n}$. Then there exists a twist E_1' of E_1 and an elliptic curve E_2 with j-invariant j_2 such that there is an $\mathbb{F}_{q^n}$-rational ℓ-isogeny $E_1' \to E_2$. Furthermore, if j_{E_1} is not equal to 0 or 1728 then we can take $E_1' = E_1$.*

Theorem 1.6. *There is an algorithm that computes φ_ℓ in a time quasi linear in its size $\widetilde{O}(\ell^3)$. Over a finite field, finding the isogenous elliptic curves (of degree ℓ) is then quasi-cubic.*

Proof.

- The complex analytic method [Eng09]: if we see $\tau \mapsto j(\tau)$ and $\tau \mapsto j(\tau/\ell)$ as a modular functions on $\mathfrak{H}$; then $\varphi_\ell(\cdot, j)$ is the minimal polynomial of $j(\cdot/\ell)$ in $\mathbb{C}(j)$. One can then recover the polynomial by computing the Fourrier coefficients of j and $j(\cdot/\ell)$ with high precision. For a quasi-linear algorithm use an evaluation interpolation approach rather than linear algebra on the Fourrier coefficients.

  This approach use the fact that

  $$\varphi_\ell(j(\tau), Y) = \prod_{g\in\Gamma/\Gamma_0(\ell} (Y - j(\ell g.\tau) = \sum c_i(\tau)Y^i \quad \text{(evaluation)}$$

  and then interpolate the coefficients $c_i(\tau)$ (which are invariant under the action of Γ) as polynomials in j (interpolation).
- The CRT method [BLS09]: use Vélu's formulas to compute $\varphi_\ell \mod p$ for small p and use the CRT to recover the full modular polynomial.

□

1.3. **Finding an isogeny between two isogenous elliptic curves.** Suppose that E_1 and E_2 are ℓ-isogenous elliptic curves, we want to compute $f : E_1 \to E_2$. The explicit forms of f is given by Vélu's formula, which give a normalized isogeny (meaning that $f^*w_{E_2} = w_{E_1}$). We first need to normalize E_2. Over $\mathbb{C}$, the equation of the normalized curve E_2 is given by the Eisenstein series $\mathcal{E}_4(\ell\tau)$ and $\mathcal{E}_6(\ell\tau)$. We have $j'(\ell\tau)/j(\ell\tau) = -\mathcal{E}_6(\tau)/\mathcal{E}_4(\tau)$. By differencing the modular polynomial, we recover the differential logarithms.

Proposition 1.7. *From $E : y^2 = x^3 + ax + b$, a normalized model of j_{E_2} is given by the Weierstrass equation*

$$y^2 = x^3 + Ax + B$$

where $A = -\frac{1}{48}\frac{J^2}{j_{E'}(j_{E'}-1728)}$, $B = -\frac{1}{864}\frac{J^3}{j_{E'}^2(j_{E'}-1728)}$ *and* $J = -\frac{18}{\ell}\frac{b}{a}\frac{\varphi_\ell'^{(X)}(j_E,j_{E'})}{\varphi_\ell'^{(Y)}(j_E,j_{E'})}j_E$.

Remark 1.8. $\mathcal{E}_2(\tau)$ is the differential logarithm of the discriminant. Similar methods allow to recover $\mathcal{E}_2(\ell\tau)$, and from it $\sigma = \sum_{P\in K\setminus\{0_E\}} x(K)$.

Finding the isogeny between the normalized models (I: Stark's method). We need to find the rational function $I(x) = g(x)/h(x)$ giving the isogeny $f : (x, y) \mapsto (I(x), yI'(x))$ between E_1 and E_2. Over $\mathbb{C}$ the coordinates of the elliptic curve are given by the elliptic functions: $x = \wp(z)$ and $y = \wp'(z)$. We have to find I such that $\wp_{E_2}(z) = I \circ \wp_{E_1}(z)$. Stark's idea is to develop $\wp_{E_2}$ as a continuous fraction in $\wp_{E_1}$, and approximate I as p_n/q_n. This algorithm is quasi-quadratic ($\widetilde{O}(\ell^2)$).
Finding the isogeny between the normalized models (II: Elkie's method [Elk92]). Plugging f into the equation of E_2 shows that I satisfy the differential equation

$$(x^3 + ax + b)I'(x)^2 = I(x)^3 + AI(x) + B.$$

Using an asymptotically fast algorithm to solve this equation yields $I(x)$ in time quasi-linear ($\widetilde{O}(\ell)$). (Knowing σ gains a logarithmic factor.)

Algorithm 1.9. To summarize, we have the following algorithm to find an isogeny from E_1 in large characteristic [BMS+08] in time $\widetilde{O}(\ell^3 + \ell\log^2 q)$:

(1) Compute φ_ℓ (cost $\widetilde{O}(\ell^3)$)
(2) Specialize on j_E to obtain $\varphi_\ell(X, j_E)$ (cost $\widetilde{O}(\ell^2 \log q)$)
(3) Find a root $j_{E'}$ of $\varphi_\ell(X, j_E)$ to obtain the j-invariant of a ℓ-isogenous curve E' (cost $\widetilde{O}(\ell \log^2 q)$).
(4) Compute the normalized model for E' (cost $\widetilde{O}(\ell^2 \log q)$).
(5) Solve the differential equation (cost $\widetilde{O}(\ell \log q)$).

2. Abelian varieties

2.1. **Theta functions.** Let $A/\mathbb{C} = \mathbb{C}^g/(\mathbb{Z}^g + \Omega\mathbb{Z}^g)$ be a principally polarised abelian variety, with $\Omega \in \mathfrak{H}_g$. Recall the definition of the theta functions with characteristics $a, b \in \mathbb{Q}^g$:

$$\vartheta\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right](z, \Omega) = \sum_{n\in\mathbb{Z}^g} e^{\pi i\, {}^t(n+a)\cdot\Omega\cdot(n+a)+2\pi i\, {}^t(n+a)\cdot(z+b)}.$$

If $\mathcal{L} = \mathcal{L}_0^n$ is the polarisation of level n associated to the principal symmetric line bundle $\mathcal{L}_0$ coming from Ω, we let

$$\vartheta_i^{\mathcal{L}}(z) = \vartheta\left[\begin{smallmatrix} 0 \\ i/n \end{smallmatrix}\right](z, \Omega/n),$$

for $i \in Z(\overline{n}) = (\mathbb{Z}/n\mathbb{Z})^g$. This form a basis of the sections of $\mathcal{L}$, that is of functions f on $\mathbb{C}^g$ that satisfy the following automorphic conditions:

$$f(z + m) = f(z),$$
$$f(z + \Omega m) = e^{-\pi i n\, {}^t m\cdot\Omega\cdot m - 2\pi i n\, {}^t z\cdot m} f(z).$$

Furthermore, this is the unique basis (up to multiplication by a constant) such that translation by a point of n-torsion is given by

$$\vartheta_b(z + \frac{m_1}{n} + \frac{\Omega m_2}{n}) = e^{-\pi i\, {}^t m_2\cdot\frac{\Omega}{n}\cdot m_2 - 2\pi i\, {}^t m_2\cdot z} e^{-2\pi i\, {}^t b\cdot m_2}\vartheta_{b+m_1}(z),$$

for $m_1, m_2 \in \mathbb{Z}^g$ (for more details on the canonical choice of a basis of sections, see [Mum83; Mum66]).

Proposition 2.1 (Lefschetz)**.**

- *If $n \geq 3$ we get an embedding of A into projective space;*
- *If $n = 2$ and $\mathcal{L}_0$ is indecomposable, we get an embedding of the Kummer variety $A/\pm 1$;*

- $(A, \mathcal{L}, A[n])$ *is entirely determined by the theta null point* $(\vartheta_i(0))_{i\in Z(\overline{n})}$ *when* $2 \mid n$ *and* $n \geq 4$. *(In fact the theta null point determines a symmetric theta structure of level* n *on* A*).*

We now suppose that $2 \mid n$, so $\mathcal{L}$ is totally symmetric.

Theorem 2.2 (Riemann relations)**.** *Let* $x_1, y_1, u_1, v_1, z \in \mathbb{C}^g$, *such that* $2z = x_1 + y_1 + u_1 + v_1$ *and let* $x_2 = z - x_1$, $y_2 = z - y_1$, $u_2 = z - u_1$, $v_2 = z - v_1$. *Then for all characters* $\chi \in \hat{Z}(\overline{2})$ *and all* $i, j, k, l, m \in Z(\overline{n})$ *such that* $i+j+k+l = 2m$, *if* $i' = m-i$, $j' = m-j$, $k' = m-k$ *and* $l' = m-l$, *then*

$$\Big(\sum_{t\in Z(\overline{2})} \chi(t)\vartheta_{i+t}(x_1)\vartheta_{j+t}(y_1)\Big).\Big(\sum_{t\in Z(\overline{2})} \chi(t)\vartheta_{k+t}(u_1)\vartheta_{l+t}(v_1)\Big) = \Big(\sum_{t\in Z(\overline{2})} \chi(t)\vartheta_{i'+t}(x_2)\vartheta_{j'+t}(y_2)\Big).\Big(\sum_{t\in Z(\overline{2})} \chi(t)\vartheta_{k'+t}(u_2)\vartheta_{l'+t}(v_2)\Big).$$

In particular, we have the addition formulae for $z_1, z_2 \in \mathbb{C}^g$ *(with* χ, i, j, k, l *like before):*

$$\Big(\sum_{t\in Z(\overline{2})} \chi(t)\vartheta_{i+t}(z_1+z_2)\vartheta_{j+t}(z_1-z_2)\Big).\Big(\sum_{t\in Z(\overline{2})} \chi(t)\vartheta_{k+t}(0)\vartheta_{l+t}(0)\Big) = \Big(\sum_{t\in Z(\overline{2})} \chi(t)\vartheta_{-i'+t}(z_2)\vartheta_{j'+t}(z_2)\Big).\Big(\sum_{t\in Z(\overline{2})} \chi(t)\vartheta_{k'+t}(z_1)\vartheta_{l'+t}(z_1)\Big).$$

Theorem 2.3 (Moduli space)**.**

- *If* $n \geq 4$, *then the homogeneous equations determining the locus of the embedding of* A *into projective spaces are generated by Riemann relations.*
- *If* $n > 4$ *then the moduli space* $\mathcal{A}_{g,n}$ *of abelian varieties with a level* n *symmetric theta structure form an open set inside the locus determined by Riemann relations on theta null points.*

Proof. [Mum66; Mum67a; Mum67b; Kem89]. □

2.2. **Isogeny from the kernel.**

Theorem 2.4 (Isogeny theorem)**.** *Let* $f : A = \mathbb{C}^g/(\mathbb{Z}^g \oplus \Omega\mathbb{Z}^g) \to B = \mathbb{C}^g/(\mathbb{Z}^g \oplus \frac{1}{\ell}\Omega\mathbb{Z}^g) : z \mapsto z$ *be the canonical isogeny with kernel* $K = \frac{1}{\ell}\Omega\mathbb{Z}^g/\Omega\mathbb{Z}^g$. *Then if we use the basis with level* ℓn *for* A *and the basis with level* n *for* B, *we get that*

$$f^*\left(\vartheta\left[\begin{smallmatrix}0\\ b/n\end{smallmatrix}\right](z, \frac{1}{n}(\frac{\Omega}{\ell}))\right) = \vartheta\left[\begin{smallmatrix}0\\ b\ell/n\ell\end{smallmatrix}\right](z, \frac{\Omega}{n\ell})$$

ie $f^*\vartheta_i^B = \vartheta^A_{\varphi(i)}$ *where* $\varphi : Z(\overline{n}) \to Z(\overline{\ell n})$ *is the canonical injection.*

Theorem 2.5 (Koizumi)**.** *Let* $(\gamma_1, \dots, \gamma_r) \in \mathbb{Q}^r$, $(\delta_1, \dots, \delta_r) \in \mathbb{Q}^r$ *and* $F \in \mathrm{Gl}_r(\mathbb{Q})$ *be such that*

$${}^tF\begin{pmatrix}\gamma_1 & & 0\\ & \ddots & \\ 0 & & \gamma_r\end{pmatrix}F = \begin{pmatrix}\delta_1 & & 0\\ & \ddots & \\ 0 & & \delta_r\end{pmatrix}.$$

Let $(x_1, \dots, x_r) \in (\mathbb{C}^g)^r$, *and* $(y_1, \dots, y_r) = (x_1, \dots, x_r)F$. *Let* $(a_1, \dots, a_r)$ *and* $(b_1, \dots, b_r)$ *be elements of* $(\mathbb{C}^g)^r$, *and note*

$$(a'_1, \dots, a'_r) = (a_1, \dots, a_r)\,{}^tF^{-1},$$
$$(b'_1, \dots, b'_r) = (b_1, \dots, b_r)F.$$

Let d *be the index* $[\mathrm{Mat}_{g\times r}(\mathbb{Z}) + \mathrm{Mat}_{g\times r}(\mathbb{Z})\,{}^tF : \mathrm{Mat}_{g\times r}(\mathbb{Z})]$ *We have:*

(1) $$d\,\vartheta\left[\begin{smallmatrix}a_1\\ b_1\end{smallmatrix}\right](x_1, \gamma_1\Omega) \times \cdots \times \vartheta\left[\begin{smallmatrix}a_r\\ b_r\end{smallmatrix}\right](x_r, \gamma_r\Omega) = \sum \vartheta\left[\begin{smallmatrix}a'_1+\alpha_1\\ b'_1+\beta_1\end{smallmatrix}\right](y_1, \delta_1\Omega) \times \cdots \times \vartheta\left[\begin{smallmatrix}a'_r+\alpha_r\\ b'_r+\beta_r\end{smallmatrix}\right](y_r, \delta_r\Omega)$$

where the sum is over the elements α and β such that

$$\begin{aligned}\alpha &\in \mathrm{Mat}_{g\times r}(\mathbb{Z})\,{}^tF^{-1}/\left(\mathrm{Mat}_{g\times r}(\mathbb{Z})\bigcap\mathrm{Mat}_{g\times r}(\mathbb{Z})\,{}^tF^{-1}\right),\\ \beta &\in \mathrm{Mat}_{g\times r}(\mathbb{Z})F/\left(\mathrm{Mat}_{g\times r}(\mathbb{Z})\bigcap\mathrm{Mat}_{g\times r}(\mathbb{Z})F\right).\end{aligned}$$

Proof. See [Koi76; Kem89; Mum83; Mum91]. □

Corollary 2.6 (Changing level)**.** *Let $\ell = a^2+b^2$, and let $F = \begin{pmatrix} a & b \\ -b & a\end{pmatrix}$ so that ${}^tFF = \ell\,\mathrm{Id}$. The link between the theta coordinates of level n on A and the ones of level ℓn is given by*

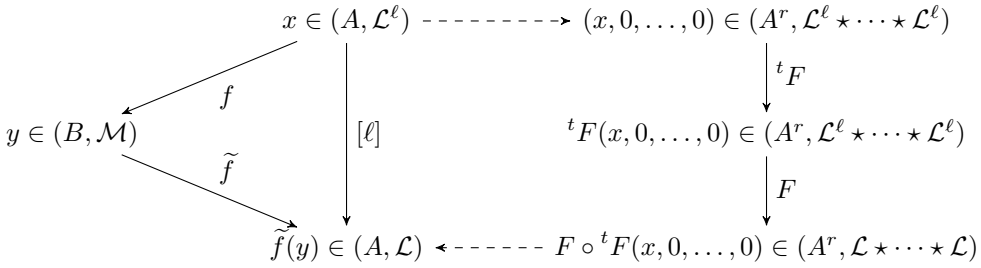
$$\vartheta^{\mathcal{L}}_{i_1}(x_1)\vartheta^{\mathcal{L}}_{i_2}(x_2) = \sum_{t\in\frac{1}{\ell}\Omega\mathbb{Z}^g/\Omega\mathbb{Z}^g} \vartheta^{\mathcal{L}^\ell}_{j_1}(y_1+at)\vartheta^{\mathcal{L}^\ell}_{j_2}(y_2+bt).$$

where $(x_1,x_2) = F(y_1,y_2)$, $(i_1,i_2) = F(j_1,j_2)$.

Theorem 2.7 (Isogeny computation)**.** *Combining the isogeny theorem and the change of level, we can compute the contragredient isogeny $\widetilde{f}:(B,\mathcal{M})\to(A,\mathcal{L})$ with kernel K while staying in level n. Let $z\in\mathbb{C}^g$, $Y=(\ell z,0,\dots,0)$ and $X = YF^{-1}$ (so that $X_1,\dots,X_r$ are integral multiples of z), let $k\in Z(\overline{n})$ and $j=(k,0,\dots,0)F^{-1}$.*

$$\vartheta^A_k(\ell z)\vartheta^A_0(0)\dots\vartheta^A_0(0) = \sum_{\substack{t_1,\dots,t_r\in K\\(t_1,\dots,t_r)F=(0,\dots,0)}} \vartheta^B_{j_1}(X_1+t_1)\dots\vartheta^B_{j_r}(X_r+t_r).$$

Proof. See [CR13] which uses the following diagram:



□

Complexity Analysis 2.8. *Let $r=1$ if ℓ is a sum of two squares, $r=2$ otherwise. Let k be the field of definition of the kernel K, and k' the field where the geometric points of K lives.*

- *From equations (in a suitable form) of K, one can compute the corresponding isogeny in time $O(\ell^{gr})$ in k [LR];*
- *From a basis of K, one can compute the corresponding isogeny in time $O(\ell^g)$ operations in k' and $O(\ell^{gr})$ operations in k.*

Proof. Let k' be the extension where the geometric points of K live.

- The isogeny formula assumes that the points are in affine coordinates. In practice, given A/k we only have projective coordinates ⇒ we use differential additions to normalize the coordinates;
- Computing the normalization factors takes $O(\log\ell)$ operations in k';
- Computing the points of the kernel via differential additions take $O(\ell^g)$ operations in k';
- If $\ell\equiv 1 \pmod 4$, applying the isogeny formula take $O(\ell^g)$ operations in k';
- If $\ell\equiv 3 \pmod 4$, applying the isogeny formula take $O(\ell^{2g})$ operations in k';

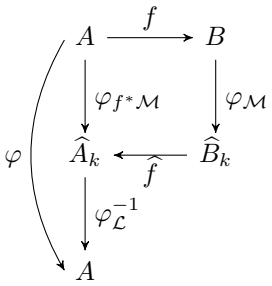
Over $\mathbb{F}_q$ the geometric points of the kernel live in a extension of degree at most $\ell^g - 1$; the total cost is then $\widetilde{O}(\ell^{2g})$ or $\widetilde{O}(\ell^{3g})$ operations in $\mathbb{F}_q$.

The complexity is much worse over a number field because we need to work with extensions of much higher degree.

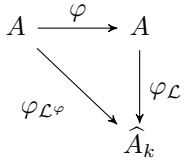
We can compute the isogeny directly given the equations (in a suitable form) of the kernel K of the isogeny, by working with "formal tuples" [LR]. When K is rational, this gives a complexity of $\widetilde{O}(\ell^g)$ or $\widetilde{O}(\ell^{2g})$ operations in $\mathbb{F}_q$. When given a basis of K, computing the equations of K costs $O(\ell^g)$ operations in k'. □

2.3. **Cyclic isogenies.** Let $f : A \to B$ be an isogeny with cyclic kernel, and assume that we have principal polarization $\mathcal{L}_0$ and $\mathcal{M}_0$ on A and B. Let $\mathcal{L} = \mathcal{L}_0^n$ and $\mathcal{M} = \mathcal{M}_0^n$.

Then there exist φ such that the following diagram commutes:



By construction, φ commutes with the Rosatti involution, so it is a (totally positive) totally real element of $\mathrm{End}^0(A)$. We note $\mathcal{L}^\varphi = f^*\mathcal{M}$ so that we have the following diagram



It is easy to see that $K = \mathrm{Ker}\, f$ is isotropic under the commutator pairing of $\mathcal{L}^\varphi$.

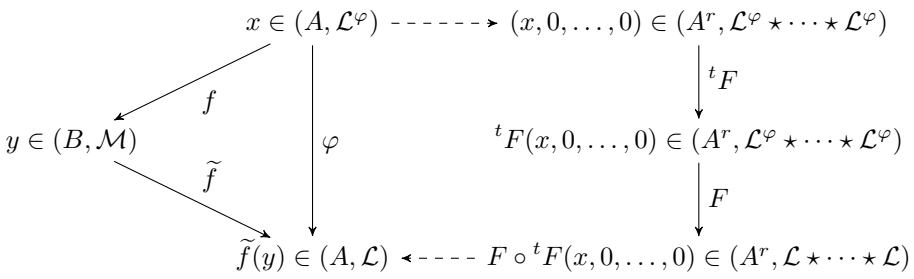
Assume that we can find a matrix $F \in \mathrm{Mat}_r(\mathrm{End}^+(A))$ such that $\varphi \,\mathrm{Id} = {}^tFF$. Then we can compute the φ-contragredient isogey $\widetilde{f}$ as follow.

Proposition 2.9. *Let $(B, \mathcal{M}_0)$ be a ppav with a symmetric theta structure on $G(\mathcal{M})$ where $\mathcal{M} = \mathcal{M}_0^n$ is of level n even. Let $K' \subset B[\ell]$ be a maximal isotropic subgroup for $\mathcal{M}^\varphi$ and $\widetilde{f} : B \to A = B/K'$ be the associated isogeny. Assume that ℓ is prime to $2n$; then the theta structure on $G(\mathcal{M})$ induces a unique polarization $\mathcal{L}$ of level n on A and a unique compatible symmetric theta structure on $G(\mathcal{L})$. Let $F \in \mathrm{Mat}_r(O_{K_0})$ be such that ${}^tFF = \varphi \,\mathrm{Id}$.*

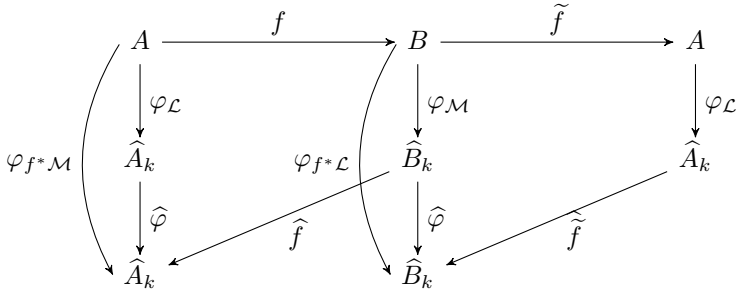
Let $i \in K_1(\mathcal{L})$ and $(j_1, \dots, j_r) \in K_1(\mathcal{M})^r$ be the unique preimage of $(i, 0, \dots, 0)$ by F. Let y be a geometric point of B and let $Y = {}^tF(y, 0, \dots, 0) \in B^r$. Then (up to a constant λ that may depend on y this time)

$$\vartheta_i^{\mathcal{L}}(\widetilde{f}(y)) \cdot \dots \cdot \vartheta_0^{\mathcal{L}}(0) = \lambda \sum_{\substack{(t_1,\dots,t_r)\in K'^r \\ F(t_1,\dots,t_r)=(0,\dots,0)}} \vartheta_{j_1}^{\mathcal{M}}(Y_1 + t_1) \cdot \dots \cdot \vartheta_{j_r}^{\mathcal{M}}(Y_r + t_r). \tag{2}$$

Proof. This is a work in progress with Dimitar Jetchev and Alina Dudeanu. We have the following diagram describing the steps of the isogeny computation



The full picture is summarized by:



□

Remark 2.10. In dimension 2, if ℓ splits completely into principal ideals as $\ell = \ell_1\ell_2$ inside the real class field, then there are two types of cyclic isogenies according to whether the kernel is inside $A[\varphi_1]$ or $A[\varphi_2]$ where we note φ_i a generator of the ideal ℓ_i.

2.4. **Moduli spaces.**

Theorem 2.11 (Duplication formulae)**.** *For all* $\chi \in \frac{1}{2}\mathbb{Z}^g/\mathbb{Z}^g$,

$$\vartheta\left[\begin{smallmatrix}\chi\\0\end{smallmatrix}\right](0, 2\frac{\Omega}{n})^2 = \frac{1}{2^g}\sum_{t\in\frac{1}{2}\mathbb{Z}^g/\mathbb{Z}^g} e^{-2i\pi 2\,{}^t\chi\cdot t}\vartheta\left[\begin{smallmatrix}0\\t\end{smallmatrix}\right](0, \frac{\Omega}{n})^2$$

$$\vartheta\left[\begin{smallmatrix}0\\i/2\end{smallmatrix}\right](0, 2\Omega)^2 = \frac{1}{2^g}\sum_{i_1+i_2=0 \pmod 2} \vartheta\left[\begin{smallmatrix}0\\i_1/2\end{smallmatrix}\right](0, \Omega)\vartheta\left[\begin{smallmatrix}0\\i_2/2\end{smallmatrix}\right](0, \Omega);$$

Example 2.12. In genus 1, via a simple change of variables, we recover the AGM:

$$\vartheta\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right](0, 2\Omega)^2 = \frac{\vartheta\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right](0, \Omega)^2 + \vartheta\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right](0, \Omega)^2}{2}$$

$$\vartheta\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right](0, 2\Omega)^2 = \sqrt{\vartheta\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right](0, \Omega)^2\vartheta\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right](0, \Omega)^2}$$

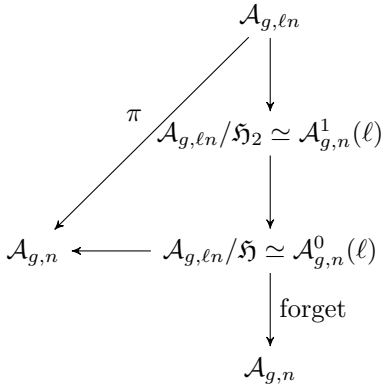
The duplication formulae allows, starting from the theta constants of level 2 of an abelian variety A_1 to compute the squares of the theta constants of level 2 of a 2-isogenous abelian variety A_2. Note that not all square roots correspond to valid theta constants, but a clever use of Riemann relations along with compatible additions from [LR13] allows to identify the "good" square roots from the "bad" ones (work in progress with David Lubicz).

Applications:

- Over $\mathbb{F}_{2^m}$, starting from an ordinary abelian variety $A = A_1$, (a subsequence of) the modular invariants of the abelian varieties A_i converge to the canonical lift $\widetilde{A}$;
- Over $\mathbb{Q}_{2^m}$, staring with $\widetilde{A}_1$, and applying the duplication formulae to get $\widetilde{A}_d$, then we get the same theta null point up to a constant u equal to the product of the eigenvalues of the Frobenius π inversible modulo 2 [Mes02].

- Over $\mathbb{C}$, we can recover the period matrix Ω (in a fundamental domain) from the theta null point when $g = 1$ or $g = 2$. Plugging a Newton iteration allows to compute theta constants in quasi-linear time from the period matrix [Dup06].

For modular correspondances, we can look at this diagram:



Assume for simplicity that n is prime to ℓ, and look at the projection map coming from the isogeny theorem $\pi : \mathcal{A}_{g,\ell n} \to \mathcal{A}_{g,n}$, $(a_i)_{i\in Z(\overline{\ell n})} \mapsto (a_i)_{i\in Z(\overline{n})}$. From a theta null point $(B, \mathcal{M}, (b_i)_{i\in Z(\overline{n})}) \in \mathcal{A}_{g,n}$, the (non degenerate) fibers in $\pi^*((b_i)_{i\in Z(\overline{n})})$ corresponds to theta null points (of level ℓn) of abelian varieties ℓ-isogenous to B with a compatible theta structure.

Theorem 2.13. *Let $\mathfrak{H}$ be the subgroup of symmetric automorphisms of the Heisenberg group of level ℓn that fix the subgroup of level n. The group $\mathfrak{H}$ is a semidirect product $\mathfrak{H}_2 \times \overline{\mathfrak{H}_1}$ which acts on the fibers as follow:*

- *$\mathfrak{H}_1$ is generated by the actions*
$$(a_i)_{i\in Z(\overline{\ell n})} \mapsto (a_{\psi(i)})_{i\in Z(\overline{\ell n})}$$
for an automorphism $\psi : Z(\overline{\ell n}) \to Z(\overline{\ell n})$ fixing $Z(\overline{n})$.
- *$\mathfrak{H}_2$ is generated by the actions*
$$(a_i)_{i\in Z(\overline{\ell n})} \mapsto (e_{\overline{\ell n}}(\psi(i), i)a_i)_{i\in Z(\overline{\ell n})}$$
where ψ is a symmetric morphism $Z(\overline{\ell n}) \to \hat{Z}(\overline{\ell})$, coming from a symmetric morphism $\psi_2 : Z(\overline{\ell}) \to \hat{Z}(\overline{\ell})$. (Where symmetric means that $\psi_2(x)(y) = \psi_2(y)(x)$.)

The fiber is reduced of dimension 0. Furthermore the action of $\mathfrak{H}$ on the geometric points in the fibers has the following properties

- *A geometric point $(a_i)_{i\in Z(\overline{\ell n})}$ in the fiber $\pi^*((b_i)_{i\in Z(\overline{n})})$ is degenerate if and only if the action of $\mathfrak{H}$ on it is non free.*
- *Two valid theta null points in the fiber correspond to the same isogenous abelian variety (with a different theta structure) if and only if they are in the same orbit under the action of $\mathfrak{H}$.*

In particular, $\mathcal{A}_{g,\ell n}/\mathfrak{H}_2$ is isomorphic to $\mathcal{A}^1_{g,n}(\ell)$, the moduli spaces classifying abelian varieties $(B, \mathcal{M})$ with a level n symmetric theta structure and a basis of a maximal isotropic kernel K in the ℓ-torsion; and In particular, $\mathcal{A}_{g,\ell n}/\mathfrak{H}_2$ is isomorphic to $\mathcal{A}^0_{g,n}(\ell)$, the moduli spaces classifying abelian varieties $(B, \mathcal{M})$ with a level n symmetric theta structure and a maximal isotropic kernel K in the ℓ-torsion.

Proof. See [FLR11], where we also give a method to construct all degenerate points in the fiber. In dimension 2, $\pi^*((b_i)_{i\in Z(\overline{n})})/\mathfrak{H}$ is of size $\ell^3 + \ell^2 + \ell + 1$ (the number of ℓ-isogenies starting from B). The size of $\mathfrak{H}_1$ is $(\ell^2 - 1)(\ell^2 - \ell)$ while the size of $\mathfrak{H}_2$ is ℓ^3, so the number of valid theta null points in the fiber is $\ell^{10} - \ell^8 - \ell^6 + \ell^4$. In dimension g we get a bound of $O(\ell^{2g^2+g})$. □

While combining Riemann relations with Koizumi's like relations allows to give equations for the moduli space $\mathcal{A}^1_{g,n}(\ell)$ (ongoing work with David Lubicz), for isogenies computations we want equations

of the moduli space $\mathcal{A}^0_{g,n}(\ell)$, or more precisely in its projection inside $\mathcal{A}_{g,n} \times \mathcal{A}_{g,n}$. Furthermore, for practical applications we want these equations to be in lexicographical Grőbner basis.

Remark 2.14. The equations for $\mathcal{A}^1_{g,n}(\ell)$ that we have allow, starting from two isogenous abelian varieties to recover the basis of the corresponding kernel by solving a Grőbner system. (Because when we have a point in this intermediate fiber, it is straightforward to recover a geometric point in the fiber $\pi^*((b_i)_{i\in Z(\overline{n})})$ and from it a basis of the kernel).

Theorem 2.15. *In dimension 2, let $(b_i)_{i\in\{1,2,3\}}$ be modular invariants on $\mathcal{A}_2$ (the moduli space of principally polarized abelian surfaces).*

Then an evaluation-interpolation algorithm can compute (in time quasi-linear in the output) the modular polynomials

$$\begin{aligned}\varphi_1(b_1,b_2,b_3,b_1') &= 0\\ b_2'\varphi_1(b_1,b_2,b_3,b_1') &= \psi_2(b_1,b_2,b_3,b_1')\\ b_3'\varphi_1(b_1,b_2,b_3,b_1') &= \psi_3(b_1,b_2,b_3,b_1')\end{aligned}$$

classifying the couple of invariants $(b_i),(b_i')$ of ℓ-isogenous abelian surfaces.

Here the modular polynomials are actually rational functions, where the denominator lies in $\mathbb{Q}(b_1,b_2,b_3)$ and comes from the Humber surface of discriminant ℓ^2 that classify abelian surfaces ℓ-isogenous to a product of elliptic curves (with the product polarisation).

Proof. See [Dup06] which uses Igusa invariants and computed the modular polynomials of level 2. This work was extended by Milio which used invariants from Streng's phd thesis and computed the modular polynomials of level 3.

Instead of Igusa invariants, using quotient of level 2 theta constant allows to get much smaller polynomials with lots of symmetries. In this case we can prove that the denominator is of total degree $\ell^3-\ell$. For more details we refer to an upcoming article by Milio. Milio also computed these polynomials up to level 7.

For instance in the evaluation we have that

$$\varphi_1(b_i(\tau)),Y) = \prod_{g\in\Gamma_{2,4}/\Gamma_{2,4}\bigcap\Gamma_0(\ell)} (Y - b_i(\ell g.\tau)),$$

where $b_i(\tau)=\vartheta_i(\tau)/\vartheta_0(\tau)$. For the evaluation we use the fact that theta constant of levels 2 generate the field of modular functions (of weight 0) invariant under $\Gamma_{2,4}$. □

References

[BMS+08] A. Bostan, F. Morain, B. Salvy, and E. Schost. "Fast algorithms for computing isogenies between elliptic curves". In: *Mathematics of Computation* 77.263 (2008), pp. 1755–1778 (cit. on p. 3).

[BLS09] R. Bröker, K. Lauter, and A. Sutherland. *Modular polynomials via isogeny volcanoes*. 2009. arXiv: 1001.0402 (cit. on p. 2).

[CR13] R. Cosset and D. Robert. "An algorithm for computing (ℓ,ℓ)-isogenies in polynomial time on Jacobians of hyperelliptic curves of genus 2". Accepted for publication at Mathematics of computation. Oct. 2013. URL: http://www.normalesup.org/~robert/pro/publications/articles/niveau.pdf. HAL: hal-00578991, eprint: 2011/143 (cit. on p. 5).

[Dup06] R. Dupont. "Moyenne arithmetico-geometrique, suites de Borchardt et applications". In: *These de doctorat, Ecole polytechnique, Palaiseau* (2006) (cit. on pp. 8, 9).

[Elk92] N. Elkies. "Explicit isogenies". In: *manuscript, Boston MA* (1992) (cit. on p. 3).

[Eng09] A. Enge. "Computing modular polynomials in quasi-linear time". In: *Math. Comp* 78.267 (2009), pp. 1809–1824 (cit. on p. 2).

[FLR11] J.-C. Faugère, D. Lubicz, and D. Robert. "Computing modular correspondences for abelian varieties". In: *Journal of Algebra* 343.1 (Oct. 2011), pp. 248–277. DOI: 10.1016/j.jalgebra.2011.06.031. arXiv: 0910.4668 [cs.SC]. URL: http://www.normalesup.org/~robert/pro/publications/articles/modular.pdf. HAL: hal-00426338 (cit. on p. 8).

[Kem89] G. Kempf. "Linear systems on abelian varieties". In: *American Journal of Mathematics* 111.1 (1989), pp. 65–94 (cit. on pp. 4, 5).

[Koh96] D. Kohel. "Endomorphism rings of elliptic curves over finite fields". PhD thesis. University of California, 1996 (cit. on p. 1).

[Koh03] D. Kohel. "The AGM-$X_0(N)$ Heegner point lifting algorithm and elliptic curve point counting". In: *Advances in cryptology—ASIACRYPT 2003*. Vol. 2894. Lecture Notes in Comput. Sci. Berlin: Springer, 2003, pp. 124–136 (cit. on p. 2).

[Koi76] S. Koizumi. "Theta relations and projective normality of abelian varieties". In: *American Journal of Mathematics* (1976), pp. 865–889 (cit. on p. 5).

[LR] D. Lubicz and D. Robert. "Computing separable isogenies in quasi-optimal time" (cit. on pp. 5, 6).

[LR13] D. Lubicz and D. Robert. "A generalisation of Miller's algorithm and applications to pairing computations on abelian varieties". Mar. 2013. URL: http://www.normalesup.org/~robert/pro/publications/articles/optimal.pdf. HAL: hal-00806923, eprint: 2013/192 (cit. on p. 7).

[Mes02] J.-F. Mestre. *Notes of a talk given at the Cryptography Seminar Rennes*. 2002. URL: http://www.math.univ-rennes1.fr/crypto/2001-02/mestre.ps (cit. on p. 7).

[Mum66] D. Mumford. "On the equations defining abelian varieties. I". In: *Invent. Math.* 1 (1966), pp. 287–354 (cit. on pp. 3, 4).

[Mum67a] D. Mumford. "On the equations defining abelian varieties. II". In: *Invent. Math.* 3 (1967), pp. 75–135 (cit. on p. 4).

[Mum67b] D. Mumford. "On the equations defining abelian varieties. III". In: *Invent. Math.* 3 (1967), pp. 215–244 (cit. on p. 4).

[Mum83] D. Mumford. *Tata lectures on theta I*. Vol. 28. Progress in Mathematics. With the assistance of C. Musili, M. Nori, E. Previato and M. Stillman. Boston, MA: Birkhäuser Boston Inc., 1983, pp. xiii+235. ISBN: 3-7643-3109-7 (cit. on pp. 3, 5).

[Mum91] D. Mumford. *Tata lectures on theta III*. Vol. 97. Progress in Mathematics. With the collaboration of Madhav Nori and Peter Norman. Boston, MA: Birkhäuser Boston Inc., 1991, pp. viii+202. ISBN: 0-8176-3440-1 (cit. on p. 5).

[Vél71] J. Vélu. "Isogénies entre courbes elliptiques". In: *Compte Rendu Académie Sciences Paris Série A-B* 273 (1971), A238–A241 (cit. on p. 1).

INRIA Bordeaux–Sud-Ouest, 200 avenue de la Vieille Tour, 33405 Talence Cedex FRANCE
E-mail address: damien.robert@inria.fr
URL: http://www.normalesup.org/~robert/

4 ARITHMETIC ON ABELIAN AND KUMMER VARIETIES (2014-12)

These notes are available at http://www.normalesup.org/~robert/pro/publications/notes/2015-05-Bordeaux-Arithmetic.pdf.

They give an introduction to the results of [LR16].

Arithmetic on Abelian and Kummer varieties

Notes of a talk given for the Lfant Algorithmic Number Theory Seminar — Bordeaux.
Based on earlier talks given in Grenoble and Caen.

Abstract. In this talk we give an outline of the results obtained in [LR14]. The first part is a review of the arithmetic on elliptic curves and Jacobians of hyperelliptic curves. The second part is more sophisticated and review the algebraic theory of theta functions, and the multiplication map. The much more elementary third part use the geometric results from the second one to improve the arithmetic on Abelian and Kummer varieties. **Warning:** These notes are in a very rough state, and probably contain a lot of errors, refer to the article for more details! Also the cost of the arithmetic mentioned for the different models do not always count the same thing, sometime we forget multiplication by small constants and sometime look at the addition with a normalized projective point, so be careful before comparing them!

Contents

1. Arithmetic on Elliptic Curves

Elliptic curve in short Weierstrass form over a field k $E : y^2 = x^3 + ax + b$ (always such a model when $\operatorname{char} k > 3$).

- Distinct points P and Q:

$$P + Q = -R = (x_R, -y_R)$$
$$\lambda = \frac{y_Q - y_P}{x_Q - x_P}$$
$$x_R = \lambda^2 - x_P - x_Q$$
$$y_R = y_P + \lambda(x_R - x_P)$$

Date: 2014-12-17.

(If $x_P = x_Q$ then $P = -Q$ and $P + Q = 0_E$).

- If $P = Q$, then λ comes from the tangent at P:

$$\lambda = \frac{3x_P^2 + b}{2y_P}$$
$$x_R = \lambda^2 - 2x_P$$
$$y_R = y_P + \lambda(x_R - x_P)$$

One can avoid divisions by working with projective coordinates $(X : Y : Z)$:

$$E : Y^2Z = X^3 + aXZ^2 + bZ^3.$$

Cost for an addition: 11M+7S in Extended Jacobian coordinates (not counting multiplication by small constants).

The scalar multiplication $P \mapsto n.P$ is computed via the standard double and add algorithm, on average $\log n$ doubling and $1/2 \log n$ additions. Standard tricks to speed-up include NAF form, windowing…The multiscalar multiplication $(P, Q) \mapsto n.P + m.Q$ can also be computed via doubling and the addition of P, Q or $P + Q$ according to the bits of n and m, on average $\log N$ doubling and $3/4 \log N$ additions where $N = \max(n, m)$. GLV idea: if there exists an efficiently computable endomorphism α such that $\alpha(P) = u.P$ where $u \approx \sqrt{n}$, then replace the scalar multiplication $n.P$ by the multiscalar multiplication $n_1P + n_2\alpha(P)$. One can expect n_1 and n_2 to be half the size of $n \Rightarrow$ from $\log n$ doubling and $1/2 \log n$ additions to $1/2 \log n$ doubling and $3/8 \log n$ additions.

Edwards curves: $E : x^2 + y^2 = 1 + dx^2y^2$, $d \neq 0, -1$, $\operatorname{char} k > 2$. Addition of $P = (x_1, y_1)$ and $Q = (x_2, y_2)$:

$$P + Q = \left(\frac{x_1y_2 + x_2y_1}{1 + dx_1x_2y_1y_2}, \frac{y_1y_2 - x_1x_2}{1 - dx_1x_2y_1y_2} \right)$$

Neutral element: $(0, 1)$; $-(x, y) = (x, y)$; $T = (1, 0)$ has order 4, $2T = (0, 1)$. (Conversely every elliptic curve with a point of 4-torsion has an Edwards curve model). When $d = 0$ we get a circle (a curve of genus 0) and we find back the addition law on the circle coming from the sine and cosine laws. If d is not a square in K, then there are no exceptional points: the denominators are always nonzero (for rational points in K) so we have a complete addition law (very useful to prevent some Side Channel Attacks). Cost for an addition: 10M+1S (Projective coordinates), 9M+1S (Inverted coordinates).

Twisted Edwards curves: $E : ax^2 + y^2 = 1 + dx^2y^2$. Addition of $P = (x_1, y_1)$ and $Q = (x_2, y_2)$:

$$P + Q = \left(\frac{x_1y_2 + x_2y_1}{1 + dx_1x_2y_1y_2}, \frac{y_1y_2 - ax_1x_2}{1 - dx_1x_2y_1y_2} \right)$$

Neutral element: $(0, 1)$; $-(x, y) = (x, y)$; $T = (0, -1)$ has order 2 (conversely if all points of 2-torsion of an elliptic curve E are rational then E is 2-isogenous to a twisted Edwards curve). Extensively studied by Bernstein and Lange, still complete addition if a is a square and d not a square. Cost for an addition: 10M+1S (Projective coordinates), 9M (Extended coordinates), 8M (Extended coordinates with $a = -1$).

Montgomery curves: $E : By^2 = x^3 + Ax^2 + x$ (birationally equivalent to twisted Edwards curves). The map $E \to \mathbb{A}^1, (x, y) \mapsto (x)$ maps E to the Kummer line $K_E = E/\pm 1$. We represent a point $\pm P \in K_E$ by the projective coordinates $(X : Z)$ where $x = X/Z$. Differential addition: Given $\pm P_1 = (X_1 : Z_1)$, $\pm P_2 = (X_2 : Z_2)$ and $\pm(P_1 - P_2) = (X_3 : Z_3)$; then one can compute $\pm(P_1 + P_2) = (X_4 : Z_4)$ by

$$X_4 = Z_3 \left((X_1 - Z_1)(X_2 + Z_2) + (X_1 + Z_1)(X_2 - Z_2)\right)^2$$
$$Z_4 = X_3 \left((X_1 - Z_1)(X_2 + Z_2) - (X_1 + Z_1)(X_2 - Z_2)\right)^2$$

Cost: 2M+2S for a doubling and 4M+2S for a differential addition.

Montgomery's scalar multiplication: The scalar multiplication $\pm P \mapsto \pm n.P$ can be computed through differential additions if we can construct a differential chain. If $\pm[n]P = (X_n - Z_n)$, then

$$X_{m+n} = Z_{m-n}\left((X_m - Z_m)(X_n + Z_n) + (X_m + Z_m)(X_n - Z_n)\right)^2$$
$$Z_{m+n} = X_{m-n}\left((X_m - Z_m)(X_n + Z_n) - (X_m + Z_m)(X_n - Z_n)\right)^2$$

Montgomery's ladder use the chain nP, $(n+1)P$: from $nP, (n+1)P$ the next iteration computes $2nP$, $(2n+1)P$ or $(2n+1)P$, $(2n+2)P$ via one doubling and one differential addition.

2. Jacobian of hyperelliptic curves

$H : y^2 = f(x)$, $\deg f = 2g+1$: hyperelliptic curve of genus g with a rational point at infinity. Every divisor D can be represented by a reduced divisor

$$\widetilde{D} = \sum_{i=1}^{r}(P_i) - r(\infty)$$

where $r \leqslant g$ and $P_i \neq -P_j$ for $i \neq j$. The divisor D is represented by its Mumford coordinates (u, v) where if $P_i = (x_i, y_i)$:

$$u(x) = \prod(x - x_i)$$
$$v(x_i) = y_i$$
$$\deg v < \deg u \leqslant g$$
$$u(x) \mid v(x)^2 - f(x);$$

The last condition encodes that $y - v(x)$ has multiplicity $m_i = \mathrm{v}_{P_i}(D)$ at P_i. From (u, v), D is recovered by $D = \mathrm{div}(u(x)) \wedge \mathrm{div}(v(x) - y)$.

Algorithm 2.1 (Cantor's algorithm). **Input:** $D_1 = (u_1, v_1)$, $D_2 = (u_2, v_2)$;
Output: $D = (u, v)$ such that $D \sim D_1 + D_2$;

(1) **Semireduce**: Compute the extended gcd of u_1, u_2, $v_1 + v_2$

$$d = s_1u_1 + s_2u_2 + s_3(v_1 + v_2)$$
$$u = \frac{u_1u_2}{d^2}$$
$$v = \frac{s_1u_1v_2 + s_2u_2v_1 + s_3(v_1v_2 + f)}{d} \text{ modulo } u$$

(2) **Reduce**:

$$u = \frac{f - v^2}{u} \quad \text{(Use the function } f - v^2 \text{ to reduce the current divisor)}$$
$$v = -v \text{ modulo } u$$

until $\deg u \leqslant g$.

Cost in genus 2: 32M + 7S for a doubling and 36M + 5S for an addition in weighted coordinates [Lan05]; 21M + 12S for a doubling and 29M + 7S for an addition in Jacobian coordinates [HC14].

3. Complex abelian varieties

$A = (V/\Lambda, H)$ where V is a $\mathbb{C}$-ev of dimension g, Λ is a lattice of rank $2g$ and $E = \Im H$ is symplectic, $E(ix, iy) = E(x, y)$ and $E(\Lambda, \Lambda) \subset \mathbb{Z}$. If $\Lambda = \mathbb{Z}^g + \Omega\mathbb{Z}^g$ where $\Omega \in \mathfrak{H}_g$ (ie Ω symmetric, $\Im\Omega > 0$), Ω determines a principal polarisation $H_0 = (\Im\Omega)^{-1}$.

Definition 3.1 (Theta functions with characteristics $a, b \in \mathbb{Q}^g$).

$$\vartheta\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right](z, \Omega) = \sum_{n \in \mathbb{Z}^g} e^{\pi i\, {}^t(n+a)\cdot\Omega\cdot(n+a) + 2\pi i\, {}^t(n+a)\cdot(z+b)}.$$

	Montgomery	Level 2	Twisted Edwards (Inverted)	Jacobians coordinates
Doubling	$5M + 4S + 1m_0$	$3M + 6S + 3m_0$	$3M + 4S + 1m_0$	$3M + 5S$
Mixed Addition			$8M + 1S + 2m_0$	$7M + 6S + 1m_0$

	Mumford (Jacobian coordinates)	Level 2	Level 4
Doubling	$21M + 12S + 2m_0$	$7M + 12S + 9m_0$	$49M + 36S + 27m_0$
Mixed Addition	$29M + 7S$		

TABLE 1. Multiplication cost in dimension 1 and 2 (one step).

To get coordinates, we need a projective embedding, which corresponds to an (ample) line bundle $\mathcal{L}$. The sections of $\mathcal{L}$ correspond to functions f such that

$$f(z+\lambda) = a_{\mathcal{L}}(z,\lambda)f(z)$$

where $a_{\mathcal{L}}$ is the automorphic factor associated to $\mathcal{L}$, satisfying the cocycle condition

$$a_{\mathcal{L}}(z,\lambda_1+\lambda_2) = a_{\mathcal{L}}(z,\lambda_1)a_{\mathcal{L}}(z+\lambda_1,\lambda_2).$$

Theorem 3.2 (Appell-Humbert)**.**

$$a_{\mathcal{L}}(z,\lambda) = \chi(\lambda)e^{\pi H(z,\lambda)+\frac{\pi}{2}H(\lambda,\lambda)}$$

where $\chi(\lambda) = \pm 1$ (when $\mathcal{L}$ is symmetric).

If $\mathcal{L} = \mathcal{L}_0^n$ ie if the polarisation H is nH_0, the sections are called theta functions of level n. If $n = n_1n_2$ a basis is given by $\vartheta \begin{bmatrix} a/n_1 \\ b/n_2 \end{bmatrix} (n_1 z, \frac{n_1}{n_2}\Omega)$. A choice of basis is uniquely determined (up to a constant) by a representation of the action by translation by points of n-torsions.

Proposition 3.3 (Lefschetz)**.**

- *If $n \geqslant 3$ we get an embedding of A into projective space;*
- *If $n = 2$ and $\mathcal{L}_0$ is indecomposable, we get an embedding of the Kummer variety $A/\pm 1$;*

Example 3.4. Let E_1 and E_2 be two elliptic curves, $\mathcal{L}_1$ and $\mathcal{L}_2$ be the corresponding canonical polarisation coming from 0_{E_i} and Let $\mathcal{L}_0 = \mathcal{L}_1 \star \mathcal{L}_2$ be the product polarisation on $E_1 \times E_2$. Then the embedding given by the sections of $\mathcal{L} = \mathcal{L}_0^2$ give a projective embedding of $E_1/\pm 1 \times E_2/\pm 1$ which is a quotient of the Kummer variety $((E_1 \times E_2)/\pm 1)$. (Note: some terminology call the Kummer variety the quotient of A by all the automorphisms; here we only quotient by ± 1. Generically this give the same definitions but not always as the example of product varieties show).

4. Heisenberg group

$(A,\mathcal{L})/k$ polarised abelian variety over an algebraically closed field k. Assume for simplicity that $\mathcal{L}$ is ample, and $\mathcal{L}$=$\mathcal{L}_0^n$ where $\mathcal{L}_0$ is principal and n is prime to the characteristic of k.

We note $\Phi_{\mathcal{L}} : A \to \widehat{A}_k, x \mapsto \tau_x^*\mathcal{L} \otimes \mathcal{L}^{-1}$ the corresponding polarisation. The kernel $K(\mathcal{L})$ of $\Phi_{\mathcal{L}}$ is then $A[n]$.

Theta group:

- $G(\mathcal{L}) := \{(x,\varphi) \mid x \in K(\mathcal{L}), \varphi : \mathcal{L} \xrightarrow{\sim} \tau_x^*\mathcal{L}\}$.
- Group law: $(y,\psi).(x,\varphi) = (x+y, \tau_x^*\psi \circ \varphi)$:

$$\mathcal{L} \xrightarrow{\varphi} \tau_x^*\mathcal{L} \xrightarrow{\tau_x^*\psi} \tau_y^*\tau_x^*\mathcal{L}.$$

- The theta group fits into the exact sequence

$$0 \longrightarrow k^* \longrightarrow G(\mathcal{L}) \longrightarrow K(\mathcal{L}) \longrightarrow 0 \,.$$

- The commutator pairing $e_{\mathcal{L}}(x,y) = \widetilde{x}\widetilde{y}\widetilde{x}^{-1}\widetilde{y} \in k^*$ is non degenerate (Weil pairing), so $G(\mathcal{L})$ is an Heisenberg group. If $\psi : K(\mathcal{L})^2 \to k^*$ is the 2-cocycle corresponding to the central extension $G(\mathcal{L})$, then $e_{\mathcal{L}}(x,y) = \frac{\psi(x,y)}{\psi(y,x)}$.
- Action of $G(\mathcal{L})$ on $\Gamma(\mathcal{L})$:

$$(x,\varphi).f = \tau_{-x}^*(\varphi(f)).$$

Standard Heisenberg group: $K(n) := (\mathbb{Z}/n\mathbb{Z})^g \oplus \widehat{(\mathbb{Z}/n\mathbb{Z})^g}$. The Heisenberg group $G(n)$ is the central extension

$$0 \longrightarrow k^* \longrightarrow G(n) \longrightarrow K(n) \longrightarrow 0$$

given by the 2-cocycle $\psi(x,y) = x_2(y_1)$. Concretely $(\alpha, x_1, x_2).(\beta, y_1, y_2) = (\alpha\beta x_2(y_1), x_1 + y_1, x_2 + y_2)$. The symplectic isomorphism $(K(n), e_n) \simeq (K(\mathcal{L}), e_{\mathcal{L}})$ extends (not uniquely in general) to an isomorphism $\Theta_{\mathcal{L}} : G(n) \xrightarrow{\sim} G(\mathcal{L})$ (Theta structure of level n).

Theorem 4.1 (Mackey)**.** *$G(n)$ has a unique irreducible representation $V(n)$ of weight 1 (ie k^* acts by the natural character). If V is a representation of weight 1, then $V = V(n)^r$ where $r = \dim_k V^{\widetilde{K}}$ and K is a maximal isotropic subgroup of $K(n)$. Moreover the action of $\widetilde{K}$ on $V(n)$ is the standard adjoint representation, so $V(n)$ has dimension n^g.*

Proof. See [Mum66; Mum91]. □

Descent: If $K \subset K(\mathcal{L})$ is isotropic, $f : A \to B = A/K$ then
level subgroup $\widetilde{K} \subset G(\mathcal{L})$ (ie a section of K) $\Leftrightarrow$ descent data of $\mathcal{L} \Leftrightarrow \mathcal{M}$ ample bundle on B such that $f^*M = \mathcal{L}$.

Theorem 4.2. *The action of $G(\mathcal{L})$ on $\Gamma(\mathcal{L})$ is irreducible.*

Proof. If $\widetilde{K}$ is maximal, by descent theory $\mathcal{L}$ descends to a principal line bundle $\mathcal{M}$ on A/K. $\Gamma(\mathcal{L})^{\widetilde{K}} = \Gamma(\mathcal{M})$ is then of dimension 1. □

In particular $\Gamma(\mathcal{L}) \curvearrowleft G(\mathcal{L})$ is isomorphic to $V(n) \curvearrowleft G(n)$ (where $G(n)$ acts by the standard action) via $\Theta_{\mathcal{L}}$.

Explicitly if we note $Z(\overline{n}) = (\mathbb{Z}/n\mathbb{Z})^g$, $V(n) = \mathrm{Hom}(Z(\overline{n}), k)$, $(\alpha, x_1, x_2).f = y \mapsto \alpha x_2(y) f(x_1 + y)$. So there exists a unique basis $(\vartheta_i)_{i \in K_1(\mathcal{L})}$ of $\Gamma(\mathcal{L})$ such that the action of $G(\mathcal{L})$ is given by

$$(\alpha, x_1, x_2).\vartheta_i = \alpha x_2(i)\vartheta_{i-x_1}.$$

(Abuse of notation: we see $G(\mathcal{L}) = k^* \times K_1(\mathcal{L}) \times K_2(\mathcal{L})$ as a set, where $K(\mathcal{L}) = K_1(\mathcal{L}) \oplus K_2(\mathcal{L})$ is the decomposition into maximal isotropic subgroups coming from $\Theta_{\mathcal{L}}$, and $x_2(i)$ is the action coming from the 2-cocycle.)

Concretely, ϑ_0 is a non trivial section in $\Gamma(\mathcal{L})^{\widetilde{K}_2(\mathcal{L})}$ and if $i \in K_1(\mathcal{L})$, $\vartheta_i = s(i).\vartheta_0$ where s is the canonical section coming from the theta structure and $\widetilde{K}_2 = s(K_2)$ is the level subgroup above K_2.

5. Riemann relations

5.1. The Isogeny theorem.

Theorem 5.1 (Isogeny Theorem)**.** *Let $f : (A, \mathcal{L}) \to (B, \mathcal{M})$ be an isogeny between polarised abelian varieties, $\mathcal{M}$ corresponds to a section $\widetilde{K} \subset G(\mathcal{L})$ of the kernel $K = \mathrm{Ker}\, f$. $G(\mathcal{M}) = \widetilde{K}^{\perp}/\widetilde{K}$ and the decomposition $K(\mathcal{L}) = K_1(\mathcal{L}) \oplus K_2(\mathcal{L})$ induces via f a decomposition $K(\mathcal{M}) = K_1(\mathcal{M}) \oplus K_2(\mathcal{M})$ (if we assume that $K = K_1 \bigcap K \oplus K_2 \bigcap K$). Likewise the theta structure on $G(\mathcal{L})$ induces a compatible theta structure on $G(\mathcal{M})$. We then have for $i \in K_1(\mathcal{L}) \bigcap K^{\perp}$ (up to a constant)*

$$\vartheta^{\mathcal{M}}_{f(i)} = \sum_{j - i \in K \bigcap K_1(\mathcal{L})} \vartheta^{\mathcal{L}}_j = \sum_{j \in K_1(\mathcal{L}), f(j) = i} \vartheta^{\mathcal{L}}_j = \textit{Trace of } \vartheta^{\mathcal{L}}_i \textit{ under the action of } \widetilde{K}.$$

5.2. **Riemann relations.** Let $\xi : A \times A \to A \times A, (x, y) \mapsto (x + y, x - y)$ be the isogeny coming from the group law, with kernel $\mathrm{diag}\, A[2]$. We now assume that $\mathcal{L}$ is totally symmetric, ie $\mathcal{L} = \mathcal{L}_0^n$ with $\mathcal{L}_0$ symmetric and $2 \mid n$. We have $\xi^*(\mathcal{L} \star \mathcal{L}) = \mathcal{L}^2 \star \mathcal{L}^2$ where $\mathcal{L} \star \mathcal{M} := p_1^*\mathcal{L} \otimes p_2^*\mathcal{M}$.

Proposition 5.2. *For the natural product theta structure, the isogeny theorem applied to ξ yields*

$$\vartheta^{\mathcal{L}}_{i+j}(x+y)\vartheta^{\mathcal{L}}_{i-j}(x-y) = \sum_{t \in K_1(\mathcal{L})[2]} \vartheta^{\mathcal{L}^2}_{i+t}\vartheta^{\mathcal{L}^2}_{j+t}.$$

This formula is easily inversible if we do a Fourier transform: for $\chi \in \hat{Z}(\overline{2})$ and $i \in Z(2n)$, let $U^{\mathcal{L}}_{\chi,i} = \sum_{t \in Z(\overline{2})} \chi(t)\vartheta^{\mathcal{L}^2}_{i+t}$. Then we obtain the duplication formulae

$$\vartheta^{\mathcal{L}}_{i+j}(x+y)\vartheta^{\mathcal{L}}_{i-j}(x-y) = \frac{1}{2^g} \sum_{\chi \in \hat{Z}(\overline{2})} U^{\mathcal{L}^2}_{\chi,i}(x)U^{\mathcal{L}^2}_{\chi,j}(y)$$

$$U^{\mathcal{L}^2}_{\chi,i}(x)U^{\mathcal{L}^2}_{\chi,j}(y) = \sum_{t \in Z(\overline{2})} \chi(t)\vartheta^{\mathcal{L}}_{i+j+t}(x+y)\vartheta^{\mathcal{L}}_{i-j+t}(x-y)$$

Remark 5.3. In term of analytic theta functions, we have $\vartheta^{\mathcal{L}}_i(z) = \vartheta\left[\begin{smallmatrix}0\\ i/l\end{smallmatrix}\right](z, \frac{\Omega}{\ell})$, $\vartheta^{\mathcal{L}^2}_i(z) = \vartheta\left[\begin{smallmatrix}0\\ i/2l\end{smallmatrix}\right](z, \frac{\Omega}{2\ell})$, $U^{\mathcal{L}^2}_{\chi,i(z)} = \vartheta\left[\begin{smallmatrix}\chi/2\\ i/l\end{smallmatrix}\right](2z, \frac{2\Omega}{\ell})$.

Theorem 5.4 (Riemann relations)**.** *Let $x_1, x_2, x_3, x_4, z \in \mathbb{C}^g$, such that $2z = x_1 + x_2 + x_3 + x_4$ and let $y_1 = z - x_1$, $y_2 = z - x_2$, $y_3 = z - x_3$, $y_4 = z - y_4$. Then for all characters $\chi \in \hat{Z}(\overline{2})$ and all $i_1, i_2, i_3, i_4, m \in Z(\overline{n})$ such that $i_1 + i_2 + i_3 + i_4 = 2m$, if $j_1 = m - i_1$, $j_2 = m - j_2$, $j_3 = m - i_3$, $j_4 = m - i_4$ then*

$$(1)\quad \Big(\sum_{t \in Z(\overline{2})} \chi(t)\vartheta_{i_1+t}(x_1)\vartheta_{i_2+t}(x_2)\Big).\Big(\sum_{t \in Z(\overline{2})} \chi(t)\vartheta_{i_3+t}(x_3)\vartheta_{i_4+t}(x_4)\Big) = \Big(\sum_{t \in Z(\overline{2})} \chi(t)\vartheta_{j_1+t}(y_1)\vartheta_{j_2+t}(y_2)\Big).\Big(\sum_{t \in Z(\overline{2})} \chi(t)\vartheta_{j_3+t}(y_3)\vartheta_{j_4+t}(y_4)\Big).$$

In particular, we have the addition formulae for $z_1, z_2 \in \mathbb{C}^g$ (with χ, i_1, i_2, i_3, i_4 like before):

$$(2)\quad \Big(\sum_{t \in Z(\overline{2})} \chi(t)\vartheta_{i_1+t}(z_1+z_2)\vartheta_{i_2+t}(z_1-z_2)\Big).\Big(\sum_{t \in Z(\overline{2})} \chi(t)\vartheta_{i_3+t}(0)\vartheta_{i_4+t}(0)\Big) = \Big(\sum_{t \in Z(\overline{2})} \chi(t)\vartheta_{j_1+t}(z_2)\vartheta_{j_2+t}(z_2)\Big).\Big(\sum_{t \in Z(\overline{2})} \chi(t)\vartheta_{j_3+t}(z_1)\vartheta_{j_4+t}(z_1)\Big).$$

Proof. Using the duplication formulae the left term of eq. (1) is equal to $U_{\chi,m_1}(z_1)U_{\chi,m_2}(z_2)U_{\chi,m_3}(z_3)U_{\chi,m_4}(z_4)$ while the right term is equal to $U_{\chi,m_1}(z_1)U_{\chi,m_4}(z_4)U_{\chi,m_3}(z_3)U_{\chi,m_2}(z_2)$ where $z_1 = \frac{x_1+x_2}{2}$, $z_2 = \frac{x_1-x_2}{2}$, $z_3 = \frac{x_3+x_4}{2}$, $z_4 = \frac{x_3-x_4}{2}$ and $m_1 = \frac{i_1+i_2}{2}$, $m_2 = \frac{i_1-i_2}{2}$, $m_3 = \frac{i_3+i_4}{2}$, $m_4 = \frac{i_3-i_4}{2}$.

The differential addition comes by plugging

$$z_1 + z_2, z_1 - z_2, 0, 0 \mid -z_2, z_2, z_1, z_1$$

another useful application is the three way affine addition with

$$z_1 + z_2 + z_3, z_1, z_2, z_3 \mid 0, z_2 + z_3, z_1 + z_3, z_1 + z_2.$$

□

Question: For χ, i_1 and i_2, we need to find i_3, i_4 such that

$$\sum \chi(t)\vartheta^{\mathcal{L}}_{i_3+t}(0)\vartheta^{\mathcal{L}}_{i_4+t}(0) = U^{\mathcal{L}^2}_{\chi,\frac{i_3+i_4}{2}}(0)U^{\mathcal{L}^2}_{\chi,\frac{i_3-i_4}{2}}(0)$$

is not null. Then by eq. (2) we can recover all $\sum_{t \in Z(\overline{2})} \chi(t)\vartheta^{\mathcal{L}}_{i_1+t}(z_1+z_2)\vartheta^{\mathcal{L}}_{i_2+t}(z_1-z_2)$ and by doing appropriate sums of characters we recover all products $\vartheta^{\mathcal{L}}_{i_1}(z_1+z_2)\vartheta^{\mathcal{L}}_{i_2}(z_1-z_2)$. This is needed for

projective addition or affine differential additions. Remark: we can translate $m_3 = \frac{i_3+i_4}{2}$ and $m_4 = \frac{i_3-i_4}{2}$ by t_1, t_2 in $2Z(2n)$.

Example 5.5. Using $n = 2$ and analytic theta functions for visibility, the duplication formulae above are given by

$$\vartheta \begin{bmatrix} 0 \\ \frac{i}{2} \end{bmatrix} (z_1 + z_2, \Omega/2)\vartheta \begin{bmatrix} 0 \\ \frac{j}{2} \end{bmatrix} (z_1 - z_2, \Omega/2) = \sum_{t \in \frac{1}{2}\mathbb{Z}^g/\mathbb{Z}^g} \vartheta \begin{bmatrix} \frac{t}{2} \\ \frac{i+j}{n} \end{bmatrix} (2z_1, \Omega)\vartheta \begin{bmatrix} \frac{t}{2} \\ \frac{i-j}{n} \end{bmatrix} (2z_2, \Omega)$$

$$\vartheta \begin{bmatrix} \chi/2 \\ i/2 \end{bmatrix} (2z_1, \Omega)\vartheta \begin{bmatrix} \chi/2 \\ j/2 \end{bmatrix} (2z_2, \Omega) =$$

$$\frac{1}{2^g} \sum_{t \in \frac{1}{2}\mathbb{Z}^g/\mathbb{Z}^g} e^{-2i\pi\, {}^t\chi \cdot t} \vartheta \begin{bmatrix} 2\chi \\ \frac{i+j}{4}+t \end{bmatrix} (z_1 + z_2, \Omega/2)\vartheta \begin{bmatrix} 0 \\ \frac{i-j}{4}+t \end{bmatrix} (z_1 - z_2, \Omega/2).$$

To compute the addition law, given χ, i_1, i_2 we need to find i_3, i_4 such that

$$\vartheta \begin{bmatrix} \frac{\chi}{2} \\ \frac{i_3+i_4}{2} \end{bmatrix} (0, \Omega)\vartheta \begin{bmatrix} \frac{\chi}{2} \\ \frac{i_3-i_4}{2} \end{bmatrix} (0, \Omega) \neq 0.$$

5.3. **Multiplication map.** Let $m : A \to A \times A, x \mapsto (x, x)$ which induces the multiplication map $m^* : \Gamma(A, \mathcal{L}) \otimes \Gamma(A, \mathcal{L}) \to \Gamma(A, \mathcal{L}^2)$.

The following diagram show that $m^* = S^*\xi^*$.

$$\begin{array}{ccc} (X, \mathcal{L}^2) & & \\ \Big\downarrow S & \searrow^{m} & \\ (X \times X, \mathcal{L}^2 \star \mathcal{L}^2) & \xrightarrow{\xi} & (X \times X, \mathcal{L} \star \mathcal{L}). \end{array}$$

By the duplication formulae, m^* is then given by $\vartheta_i^{\mathcal{L}} \otimes \vartheta_j^{\mathcal{L}} \mapsto \sum_{\chi \in \hat{Z}(\overline{2})} U_{\chi,u}^{\mathcal{L}^2} U_{\chi,v}^{\mathcal{L}^2}(0)$ for any $u, v \in Z(2n)$ such that $i = u+v$, $j = u-v$, or via a change of variable $\sum_t \chi(t)\vartheta_{u+v+t}^{\mathcal{L}}(x) \otimes \vartheta_{u-v+t}^{\mathcal{L}}(x) \mapsto U_{\chi,i}^{\mathcal{L}^2}(x) U_{\chi,j}^{\mathcal{L}^2}(0)$. So the rank of the multiplication map is closely linked to the non annulation of the theta null points.

Remark 5.6 (Even and odd theta null points)**.** If $n = 2$, $U_{\chi,i}(-x) = \chi(2i)U_{\chi,i}(x)$ for $i \in Z(\overline{4})$, equivalently $\vartheta \begin{bmatrix} a/2 \\ b/2 \end{bmatrix} (-2z, \Omega) = (-1)^{{}^t a \cdot b}\vartheta(2z, \Omega)$. There is $2^{g-1}(2^g + 1)$ even theta null points vs $2^{g-1}(2^g - 1)$ odd theta null points. Ex: $g = 1$, 3 vs 1; $g = 2$, 10 vs 6; $g = 3$, 36 vs 28.

Theorem 5.7 (Mumford-Koizumi-Kempf)**.** *$\mathcal{L}_0$ is principal symmetric.*

- *$\Gamma(A, \mathcal{L}_0^n) \otimes \Gamma(A, \mathcal{L}_0^m) \to \Gamma(A, \mathcal{L}_0^{n+m})$ is surjective when $n \geqslant 2$ and $m \geqslant 3$.*
- *$\Gamma(A, \mathcal{L}_0^{2n})^+ \otimes \Gamma(A, \mathcal{L}_0^2) \to \Gamma(A, \mathcal{L}_0^{2(n+1)})^+$ is surjective when $n \geqslant 2$. Here $\Gamma(A, \mathcal{L}_0^{2n})^+$ denotes the even sections of $\Gamma(A, \mathcal{L}_0^{2n})$. Equivalently, since $\mathcal{L}_0^{2n}$ is totally symmetric, it descends to an ample line bundle $\mathcal{M}^+$ on the Kummer variety $\mathcal{K}_A = A/\pm 1$, and $\Gamma(A, \mathcal{L}_0^{2n})^+ = \Gamma(\mathcal{K}_A, \mathcal{M}^+)$.*
- *The rank of $\Gamma(A, \mathcal{L}_0^2) \otimes \Gamma(A, \mathcal{L}_0^2) \to \Gamma(A, \mathcal{L}_0^4)^+$ is equal to the number of non null even theta null points.*

5.4. **Normal projectivity.** A line bundle $\mathcal{L}$ on a variety X is projectively normal if $\Gamma(X, \mathcal{L}^n) \otimes \Gamma(X, \mathcal{L}) \to \Gamma(X, \mathcal{L}^{n+1})$ is surjective for all n or equivalently if $S(\Gamma(X, \mathcal{L})) \twoheadrightarrow \bigoplus_{n>0} \Gamma(X, \mathcal{L}^n)$. (And so if X is normal, its projective homogeneous ring in the embedding given by $\mathcal{L}$ is normal). *Remark:* $\mathcal{L}$ is very ample iff the map above is surjective for $n \gg 0$.

Corollary 5.8.

- *If $n \geqslant 3$, $(A, \mathcal{L})$ is projectively normal, and we have a projective embedding of A;*
- *If $n = 2$, the projective embedding of $\mathcal{K}_A$ is projectively normal iff the even theta null points are not null.* ***We now assume that this is the case whenever $n = 2$.***

Example 5.9. The product of the even theta null points is null whenever A is not absolutely simple or when it is the Jacobian of an hyperelliptic curve of genus $g \geqslant 3$.

5.5. **Addition, Differential addition.** Given $\vartheta_i(x)$, $\vartheta_i(y)$ we can recover (n even)

- $\vartheta_i(x+y)\vartheta_j(x-y)$ when $n>2$ ($\Rightarrow$ projective addition, affine differential addition)
- $\kappa_{ij} := \vartheta_i(x+y)\vartheta_j(x-y) + \vartheta_j(x+y)\vartheta_i(x-y)$ if $n=2$, the "symmetric sum" ($\Rightarrow$ differential projective or affine addition).

Algorithm 5.10. Differential addition with $g=1$, $n=2$.

Input: $z_P=(x_0,x_1)$, $z_Q=(y_0,y_1)$ and $z_{P-Q}=(z_0,z_1)$ with $z_0z_1\neq 0$; $z_0=(a,b)$ and $A=2(a^2+b^2)$, $B=2(a^2-b^2)$.

Output: $z_{P+Q}=(t_0,t_1)$.

(1) $t_0' = (x_0^2+x_1^2)(y_0^2+y_2^2)/A$
(2) $t_1' = (x_0^2-x_1^2)(y_0^2-y_1^2)/B$
(3) $t_0 = (t_0'+t_1')/z_0$
(4) $t_1 = (t_0'-t_1')/z_1$

Return (t_0,t_1)

Cost: 3M+6S+3m_0 for a step of the scalar ladder, compared to 5M+4S+1m_0 for the Montgomery model. In genus 2 the cost of one step is 7M+12S+9m_0.

6. Arithmetic on Kummer varieties

We assume here that $n=2$ and that the even theta null points are non zero.

The polynomial $P_{i\alpha} := X^2 - 2\frac{\kappa_{i\alpha}}{\kappa_{\alpha\alpha}}X + \frac{\kappa_{ii}}{\kappa_{\alpha\alpha}}$ has for roots $\{\frac{\vartheta_i(x+y)}{\vartheta_\alpha(x+y)}, \frac{\vartheta_i(x-y)}{\vartheta_\alpha(x-y)}\}$. Once a root is chosen, some two by two linear equations involving the κ_{ij} and the roots allows to recover the theta coordinates of $x+y$. This gives equations for the degree two scheme $\{x+y, x-y\}$.

Lemma 6.1 (Compatible additions)**.** *Given $x,y,z,t\in A(\overline{k})$ such that $x+y=z+t$ but $x-y\neq \pm z-t$ then one can compute $x+y$ $(=z+t)$ on the Kummer (from the points on the Kummer).*

Proof. This is just the intersection of the two schemes of degree two defining $\{x\pm y\}$ and $\{z\pm t\}$; in practice this is just a gcd of two degree two polynomials. □

Proposition 6.2 (Multiway additions)**.** *Let $\pm P_0\in\mathcal{K}_A(\overline{k})$ be a point not of 2-torsion. Then from $\pm P_1,\dots,\pm P_n\in\mathcal{K}_A(\overline{k})$ and $\pm(P_0+P_1),\dots,\pm(P_0+P_n)\in\mathcal{K}_A(\overline{k})$, one can compute $\pm(P_1+\cdots+P_n)$ and $\pm(P_0+P_1+\cdots+P_n)$.*

Remark 6.3. A reformulation of the proposition is that the data of $P_0+P_i\in\mathcal{K}_A(\overline{k})$ "fixes" the sign of P_i relatively to the one of P_0, and so we can compute the additions since we have "compatible" signs.

Proof. This reduces to the case $n=2$, which uses (in the generic case) $(P_1)+(P_2)=(P_1-P_0)+(P_2+P_0)$ and $(P_0+P_1)+P_2=P_1+(P_0+P_2)$. And a verification shows that in the non generic case a direct computation is possible. □

6.1. **Multi Scalar multiplication.** To speed up the scalar multiplication $P\mapsto nP$, the GLV trick [GLV01] is to use an endomorphism α and reduces the scalar multiplication to a multi scalar multiplication $m_1P_1+m_2P_2$ (for instance if $\alpha P=tP$, fix $P_1=P$, $P_2=\alpha(P)$, and $n=m_1+tm_2$). The doubling and add method works again, with the addition being either P_1, P_2 or P_1+P_2 according to the bits of (m_1,m_2).

On the Kummer variety a Montgomery ladder $mP,(m+1)P\mapsto 2mP,(2m+1)P$ or $(2m+1)P,(2m+2)P$ computes the scalar multiplication. The two dimensional scalar multiplication uses a square $\pm(mP+nQ)$, $\pm((m+1)P+nQ)$, $\pm(mP+(n+1)Q)$, $\pm((m+1)P+(n+1)Q)$ and depending whether the current bits of (m_1,m_2) is $(0,0)$, $(1,0)$, $(0,1)$ or $(1,1)$, adds $\pm(mP+nQ)$, $\pm((m+1)P+nQ)$, $\pm(mP+(n+1)Q)$ or $\pm((m+1)P+(n+1)Q)$ to the four points. But this is not interesting, we expect to halve the length of the chain by two, but each steps is twice as costly. A better approach from [Ber06] uses a triangle.

But via the compatible additions, we just need to keep two points!

Example 6.4. Given $m_1P_1+(m_2+1)P_2$, $(m_1+1)P_1+m_2P_2$, we can compute $(2m_1+1)P_1+(2m_2+1)P_2 = (m_1P_1+(m_2+1)P_2)+(P_1) = ((m_1+1)P_1+m_2P_2)+(P_2)$.

7. Changing level

For an elliptic curve $y^2 = f(x)$, the map $(x, y) \mapsto x$ maps the elliptic curve to the Kummer line. Going back to the elliptic curve involve a square root. For abelian variety, a similar map to the Kummer is $(A, \mathcal{L}^2)$ level $4 \to (\mathcal{K}_A, \mathcal{L}^+)$ level 2 via the duplication formula. We want to go back from level 2 to level 4, using only one square root. We would also like to be able to describe a point on A using just the point on $\mathcal{K}_A$ and an extra coordinate to encode the sign, like is possible on elliptic curve (going back to the full level 4 adds a lot of coordinates). This will be described in section 8

The theta constants of level 4 on A gives the points of 4 torsion, so we have the coordinates $U^{\mathcal{L}^2}_{\chi,i}(T)$ for T a point of four torsion. The duplication formulae gives $U_{\chi,i}(x)U_{\chi,i}(0) = \sum \chi(t)\vartheta_{2i+t}(x)\vartheta_t(x)$, but $U_{\chi,i}(0) = 0$ for odd coordinates, so we don't recover all level 4 coordinates given the level 2 ones. But $0 \neq U_{\chi,0}(0) = U_{\chi,i}(T_i)$ for an (explicit) point of four torsion T. So we can use $U_{\chi,i}(x)U_{\chi,i}(T_i) = \sum \chi(t)\vartheta_{2i+t}(x + T_i)\vartheta_t(x - T_i)$.

We thus need to compute $x + T_i$ via a square roots, then we can recover all the other ones via $x + T_j = (x + T_i) + (T_i - T_j)$.

7.1. **Compressing coordinates.** Another way to descend level is via the isogeny theorem:

$$\pi(\vartheta_i(x))_{i \in Z(\overline{\ell n})} \to (\vartheta_i(x))_{i \in Z(\overline{n})}$$

is the isogeny of kernel $K_2(\mathcal{L})[\ell]$.

Proof. The isogeny sends $\mathbb{C}^g/(\mathbb{Z}^g + \Omega\mathbb{Z}^g) \to \mathbb{C}^g/(\mathbb{Z}^g + \frac{\Omega}{\ell}\mathbb{Z}^g)$. Looking at the level ℓn and n theta functions we indeed have for $b \in Z(\overline{n})$ $\vartheta \left[\begin{smallmatrix} 0 \\ \ell b/\ell n \end{smallmatrix}\right](z, \frac{\Omega}{\ell n}) = \vartheta \left[\begin{smallmatrix} 0 \\ b/n \end{smallmatrix}\right](z, \frac{\Omega/\ell}{n})$. □

Let $e_1, \dots, e_g$ be a basis of $K_1(\mathcal{L})$. Then from $\widetilde{\pi}(x + \sum \lambda_i e_i)$, where $\lambda_i \in \{0, \dots, \ell - 1\}$ we can recover x (here $\widetilde{\pi}$ is the affine lift of π).

Example 7.1. $g = 1$, $\ell = 3$, $n = 2$. $\widetilde{\pi}(x_0, \dots, x_5) = (x_0, x_3)$. $x + e_1 = (x_1, \dots, x_5, x_0)$ so $\widetilde{\pi}(x + e_1) = (x_1, x_4)$ and $\widetilde{\pi}(x + 2e_1) = (x_2, x_5)$.

But $\widetilde{\pi}(x + \sum \lambda_i e_i) = \widetilde{\pi}(x) + \sum \lambda_i \widetilde{\pi}(e_i)$ so we can recover everything using multiway affine additions (which are just a composition of differential and three way affine additions).

Corollary 7.2.

- *0 is uniquely determined by $\widetilde{\pi}(0)$, $\widetilde{\pi}(e_i)$ and $\widetilde{\pi}(e_i + e_j)$ ($(1 + g + g(g + 1)/2)n^g$ coordinates).*
- *x is uniquely determined by $\widetilde{\pi}(x)$, $\widetilde{\pi}(x + e_i)$ ($(1 + g)n^g$ coordinates).*

8. Arithmetic on abelian varieties

Level (2, 4): this gives an embedding of A (if A is absolutely simple), and the compression of coordinates from above show that we can use the coordinates $\widetilde{\pi}(x), \widetilde{\pi}(x + T) = \widetilde{\pi}(x) + \widetilde{\pi}(T)$ where T is of 4-torsion.

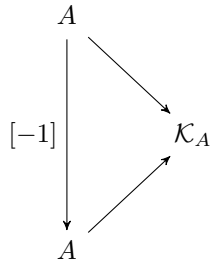
More generally, for $T \in A(\overline{k})$ such that $2T \neq 0$, we represent $x \in A(\overline{k})$ by $x \in \mathcal{K}_A(x)$, $x + T \in \mathcal{K}_A$. Addition: $(x, x + T) + (y, y + T) = (x + y = (x + T) + (y - T), x + y + T)$ (this is a three way addition and a compatible addition on the Kummer so this is quite costly). Doubling is just a doubling and a differential addition on the Kummer so this is a lot less costly.

The standard scalar multiplication costs too much because of the additions. One can instead do a Montgomery scalar multiplication with $(nx, (n + 1)x, (n + 1)x + T)$ which uses a doubling and two differential additions on the Kummer at each step.

Even better, just do a Montgomery scalar multiplication $(nx, (n + 1)x)$ on the Kummer and at the last step compute $(n + 1)x + T = nx + (x + T)$. This also works for multi-exponentiation.

Finally this representation is very compact, $x + T$ is simply represented by a root of the polynomial $P_{i\alpha}$. So we have a representation that only needs one extra coordinate compared to the Kummer one, and has a scalar multiplication (almost) as efficient, but we can still compute additions.

Remark 8.1. Changing representation: $(x, x + T_1) \mapsto (x, x + T_2)$ via $x + T_2 = (x + T_1) + (T_2 - T_1)$. This needs a choice of $T_1 + T_2$ in $\{\pm T_1 \pm T_2\}$, but this choice is necessary since $[-1]$ is an automorphism.



9. Formulae

Let $(a_i)_{i\in Z(\overline{2})}$ be the level two theta null point representing a Kummer variety $\mathcal{K}_A$ of dimension 2. Let $x=(x_i)_{i\in Z(\overline{2})}$ and $y=(y_i)_{i\in Z(\overline{2})}$, we let $X=x+y$ and $Y=x-y$. We will give formulae for the coordinates $2\kappa_{ij}=X_iY_j+X_jY_i$.

Let $i\in Z(\overline{2}), \chi\in\hat{Z}(\overline{2})$ and let

$$z_i^\chi=\big(\sum_{t\in Z(\overline{2})}\chi(t)x_{i+t}x_t\big)\big(\sum_{t\in Z(\overline{2})}\chi(t)y_{i+t}y_t\big)/\big(\sum_{t\in Z(\overline{2})}\chi(t)a_{i+t}a_t\big).$$

$\sum_t\chi(t)a_{i+t}a_t$ is simply the classical theta null point $\vartheta\begin{bmatrix}\chi/2\\ i/2\end{bmatrix}(0,\Omega)^2$. Then theorem 5.4 gives

$$\begin{aligned}
4X_{00}Y_{00}&=z_{00}^{00}+z_{00}^{01}+z_{00}^{10}+z_{00}^{11};\\
4X_{01}Y_{01}&=z_{00}^{00}-z_{00}^{01}+z_{00}^{10}+z_{00}^{11};\\
4X_{10}Y_{10}&=z_{00}^{00}+z_{00}^{01}-z_{00}^{10}-z_{00}^{11};\\
4X_{11}Y_{11}&=z_{00}^{00}-z_{00}^{01}-z_{00}^{10}+z_{00}^{11};\\
\\
2(X_{10}Y_{00}+X_{00}Y_{10})&=z_{10}^{00}+z_{10}^{01};\\
2(X_{11}Y_{01}+X_{01}Y_{11})&=z_{10}^{00}-z_{10}^{01};\\
2(X_{01}Y_{00}+X_{00}Y_{01})&=z_{01}^{00}+z_{01}^{10};\\
2(X_{11}Y_{10}+X_{10}Y_{11})&=z_{01}^{00}-z_{01}^{10};\\
2(X_{11}Y_{00}+X_{00}Y_{11})&=z_{11}^{00}+z_{11}^{11};\\
2(X_{01}Y_{10}+X_{10}Y_{01})&=z_{11}^{00}-z_{11}^{11};
\end{aligned}$$

We describe the degree two scheme $\{X,Y\}$ by the polynomial $\mathfrak{P}_\alpha(Z)=Z^2-2\frac{\kappa_{\alpha0}}{\kappa_{00}}Z+\frac{\kappa_{\alpha\alpha}}{\kappa_{00}}$ whose roots are $\{\frac{X_\alpha}{X_0},\frac{Y_\alpha}{Y_0}\}$ (where α is such that $X_\alpha Y_0-X_0Y_\alpha\neq0$). To compute κ_{00} and $\kappa_{\alpha\alpha}$ we need $4M+8S+3M_0$, and to compute $\kappa_{\alpha0}$ we need $2M+4S+2M_0$; so in total to compute $\mathfrak{P}_\alpha$, we need $6M+12S+5M_0+2I$.

Once we have a root Z, if we let $Z'=2\frac{\kappa_{\alpha0}}{\kappa_{00}}-Z$ be the conjugate root (corresponding to $\frac{Y_\alpha}{Y_0}$), we can recover the coordinates X_i, Y_i by solving the equation

$$\begin{pmatrix}1&1\\ Z&Z'\end{pmatrix}\begin{pmatrix}Y_i/Y_0\\ X_i/X_0\end{pmatrix}=\begin{pmatrix}2\kappa_{0i}/\kappa_{00}\\ 2\kappa_{\alpha i}/\kappa_{00}\end{pmatrix};$$

We find $X_i=\frac{2(Z\kappa_{0i}-\kappa_{\alpha i})}{\kappa_{00}(Z-Z')}=\frac{Z\kappa_{0i}-\kappa_{\alpha i}}{Z\kappa_{00}-\kappa_{\alpha0}}$ for $i\neq0,\alpha$ (here we have $X_0=1$, $X_\alpha=Z$). But usually we will express $Z=(X_0:X_\alpha)\in\mathbb{P}^1$ as a point in the projective line, and we find that

$$X_i=\frac{X_\alpha\kappa_{0i}-X_0\kappa_{\alpha i}}{X_\alpha\kappa_{00}-X_0\kappa_{\alpha0}}.$$

Recovering the projective coordinates of X then costs $8M$ (given the κ_{ij}). To sum up, given $Z=(X_0:X_\alpha)$ recovering X costs in total $(10M+20S+9M_0)+8M=18M+20S+9M_0$.

For a compatible addition, where $x+y=z+t$, we can find Z as the common root between $\mathfrak{P}_\alpha$ and the similar polynomial ${\mathfrak{P}'}_\alpha(Z) = Z^2 - 2\frac{\kappa'_{\alpha 0}}{\kappa'_{00}}Z + \frac{\kappa'_{\alpha\alpha}}{\kappa'_{00}}$ coming from the symmetric coordinates $z_i t_j + t_i z_j$. Computing the coefficients needed for ${\mathfrak{P}'}_\alpha$ costs $6M+12S+5M_0$. The common root is

$$Z = \frac{\frac{\kappa'_{\alpha\alpha}}{\kappa'_{00}} - \frac{\kappa_{\alpha\alpha}}{\kappa_{00}}}{-2\frac{\kappa_{\alpha 0}}{\kappa 00} + 2\frac{\kappa'_{\alpha 0}}{\kappa'_{00}}} = \frac{\kappa'_{\alpha\alpha}\kappa_{00} - \kappa_{\alpha\alpha}\kappa'_{00}}{2(\kappa'_{\alpha 0}\kappa_{00} - \kappa_{\alpha 0}\kappa'_{00})}.$$

Computing Z projectively costs $4M$. In the end, a compatible addition costs $(18M+20S+9M_0)+(6M+12S+5M_0)+4M = 28M+32S+14M_0$.

References

[Ber06] D. J. Bernstein. "Differential addition chains". 2006. URL: http://cr.yp.to/ecdh/diffchain-20060219.pdf (cit. on p. 8).

[BL07] D. Bernstein and T. Lange. *Explicit-formulas database*. 2007. URL: http://hyperelliptic.%20org/EFD.

[GLV01] R. P. Gallant, R. J. Lambert, and S. A. Vanstone. "Faster Point Multiplication on Elliptic Curves with Efficient Endomorphisms". In: *CRYPTO*. Ed. by J. Kilian. Vol. 2139. Lecture Notes in Computer Science. Springer, 2001, pp. 190–200. ISBN: 3-540-42456-3 (cit. on p. 8).

[HC14] H. Hisil and C. Costello. "Jacobian Coordinates on Genus 2 Curves". 2014. eprint: 2014/385 (cit. on p. 3).

[Lan05] T. Lange. "Formulae for arithmetic on genus 2 hyperelliptic curves". In: *Applicable Algebra in Engineering, Communication and Computing* 15.5 (2005), pp. 295–328 (cit. on p. 3).

[LR14] D. Lubicz and D. Robert. "Arithmetic on Abelian and Kummer Varieties". June 2014. URL: http://www.normalesup.org/~robert/pro/publications/articles/arithmetic.pdf. HAL: hal-01057467, eprint: 2014/493. (Cit. on p. 1).

[Mum66] D. Mumford. "On the equations defining abelian varieties. I". In: *Invent. Math.* 1 (1966), pp. 287–354 (cit. on p. 5).

[Mum91] D. Mumford. *Tata lectures on theta III*. Vol. 97. Progress in Mathematics. With the collaboration of Madhav Nori and Peter Norman. Boston, MA: Birkhäuser Boston Inc., 1991, pp. viii+202. ISBN: 0-8176-3440-1 (cit. on p. 5).

INRIA Bordeaux–Sud-Ouest, 200 avenue de la Vieille Tour, 33405 Talence Cedex FRANCE
E-mail address: damien.robert@inria.fr
URL: http://www.normalesup.org/~robert/

5 POLARISATIONS, ISOGENIES, AND PAIRINGS ON ABELIAN VARIETIES (2022-09)

These completely unfinished notes are available at http://www.normalesup.org/~robert/pro/publications/notes/2022-09-polarisations.pdf.

They were meant to accompany my talk "Isogenies between abelian varieties – an algorithmic survey" at the Leuven isogeny days 3, whose slides are available at http://www.normalesup.org/~robert/pro/publications/slides/2022-09-Leuven-Isogenies.pdf, but I never got around to writing them.

Polarisations, isogenies, and pairings in abelian varieties

DAMIEN ROBERT

Abstract. This note is an introduction to the notion of polarisation on an abelian variety and isogenies between polarised abelian varieties.

1. Introduction

These notes are written for my talk "Isogenies between abelian varieties: an overview" for the Leiden isogeny days. These are a work in progress. The aim is to give an overview of polarisations, isogenies and their strong link with the Weil pairing. I intend to update them regularly.

For my HDR thesis [Rob21a], I wrote an (unfinished) set of notes [Rob21b] on the mathematics of abelian varieties. The aim of [Rob21b] is to cover more technical topics on abelian varieties by giving the relevant pointers to the literature [Mum70; Mil91; BL04; MGE12; MFK94; Mum83; Mum84; Mum91; BLR12; FC90]…Indeed, it would be way too ambitious (for me!) to write full proofs of each of these topics, so [Rob21b] summarizes the main results and gives references for the proofs. For now it covers abelian varieties, the basic theory of abelian schemes, degenerations and the theory of Néron's models (the lifting / canonical lift part of the theory is not yet written), and pairings (Weil, Weil-Cartier, Tate, Tate-Cartier, Tate-Lichtenbaum). Further planned topics include the study of various moduli (of abelian varieties, of curves, via theta functions, CM theory). Unfortunately it is a bit stalled.

The planned topics of these notes are more elementary: polarisations, isogenies, and their strong links with the Weil pairing. I also intend to do a comparison with the case of elliptic curves (similarity / differences when going in higher dimension), and describe how these objects behave when working over $\mathbb{C}$ (they have a nice description in terms of linear algebra / quadratic forms).

Throughout these notes we only deal with separable isogenies. In particular, when looking at N-isogenies, we implicitly restrict to the case where N is prime to the characteristic p of the base field (or $p = 0$).

2. Outline

Here are planned topics.

(1) *Complex abelian varieties*: $A = V/\Lambda$, lattices, polarisations, isogenies: explain the different aspect of polarisations: as a Hermitian form on V, a symplectic form on Λ, a morphism $A \to \hat{A}$, an algebraic class of divisors (the Apell-Humbert theorem).
Jacobians, analytic description of the Theta divisor.

(2) *Abelian varieties*: Isogenies, divisibility. The dual abelian variety. Dual isogenies. Dual abelian variety: $\widehat{A} = \mathrm{Pic}^0(A)$. Dual isogeny as pullback. Alternative interpretation

Date: September 26, 2022.

$\widehat{A} = \mathrm{Ext}^1(A, \mathbb{G}_m)$ (via Mumford's theta groups), application to define the Weil-Cartier pairing of an isogeny. The Weil pairing on $A[m] \times \widehat{A}[m]$. Compatibility of pairings with isogenies. Weil pairing on the Tate modules $T_\ell A$. Biduality, Poincaré bundle, biduality and dual isogenies behave as expected with respect to the Weil pairing.

(3) *Polarisations.* Algebraic interpretations of the different facets of polarisations in the complex analytic case (Weil pairing, divisors). Characterisation of when $\phi : A \to \hat{A}$ is a polarisation. The Néron-Severi group. Field of definition of a polarisation vs field of definition of an associated divisor, the case of $k = \mathbb{F}_q$. Polarisations and pairings, Theta group, descent. Contragredient isogeny. Product polarisations, polarisations on product. The contragredient matrix is the transpose of the matrix of contragredient isogenies. The Jacobian of a curve and its theta divisor. The special case of elliptic curves.

(4) *N-isogenies.* Link with maximal isotropic kernels. Contragredient isogeny $\tilde{f}$. Characterisation of a N-isogeny via the contragredient isogeny and via pairings.

(5) *Maximal isotropic kernels.* Elementary theory of symplectic finite abelian groups [Zhm71; PSV10]. Symplectic CRT. Maximal isotropic kernel: rank g, standard kernels.

References

[BL04] C. Birkenhake and H. Lange. *Complex abelian varieties*. Second. Vol. 302. Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]. Berlin: Springer-Verlag, 2004, pp. xii+635. ISBN: 3-540-20488-1.

[BLR12] S. Bosch, W. Lütkebohmert, and M. Raynaud. *Néron models*. Vol. 21. Springer Science & Business Media, 2012.

[FC90] G. Faltings and C.-L. Chai. *Degeneration of abelian varieties*. Ergebnisse der Mathematik und ihrer Grenzgebiete (3) 22. Springer-Verlag, Berlin, 1990.

[Mil91] J. Milne. *Abelian varieties*. 1991. URL: http://www.jmilne.org/math/CourseNotes/av.html.

[MGE12] B. Moonen, G. van der Geer, and B. Edixhoven. *Abelian varieties*. Book project, 2012. URL: https://www.math.ru.nl/~bmoonen/research.html#bookabvar.

[Mum70] D. Mumford. *Abelian varieties*. Tata Institute of Fundamental Research Studies in Mathematics, No. 5. Published for the Tata Institute of Fundamental Research, Bombay, 1970, pp. viii+242.

[Mum83] D. Mumford. *Tata lectures on theta I*. Vol. 28. Progress in Mathematics. With the assistance of C. Musili, M. Nori, E. Previato and M. Stillman. Boston, MA: Birkhäuser Boston Inc., 1983, pp. xiii+235. ISBN: 3-7643-3109-7.

[Mum84] D. Mumford. *Tata lectures on theta II*. Vol. 43. Progress in Mathematics. Jacobian theta functions and differential equations, With the collaboration of C. Musili, M. Nori, E. Previato, M. Stillman and H. Umemura. Boston, MA: Birkhäuser Boston Inc., 1984, pp. xiv+272. ISBN: 0-8176-3110-0.

[Mum91] D. Mumford. *Tata lectures on theta III*. Vol. 97. Progress in Mathematics. With the collaboration of Madhav Nori and Peter Norman. Boston, MA: Birkhäuser Boston Inc., 1991, pp. viii+202. ISBN: 0-8176-3440-1.

[MFK94] D. Mumford, J. Fogarty, and F. Kirwan. *Geometric invariant theory*. Vol. 34. Springer Science & Business Media, 1994.

[PSV10] A. Prasad, I. Shapiro, and M. Vemuri. "Locally compact abelian groups with symplectic self-duality". In: *Advances in Mathematics* 225.5 (2010), pp. 2429–2454.

[Rob21a] D. Robert. "Efficient algorithms for abelian varieties and their moduli spaces". HDR thesis. Université Bordeaux, June 2021. URL: http://www.normalesup.org/~robert/pro/publications/academic/hdr.pdf. Slides: 2021-06-HDR-Bordeaux.pdf (1h, Bordeaux).

[Rob21b] D. Robert. *General theory of abelian varieties and their moduli spaces*. Jan. 2021. URL: http://www.normalesup.org/~robert/pro/publications/books/avtheory.pdf. Draft version.

[Zhm71] E. M. Zhmud'. "Symplectic geometries over finite abelian groups". In: *Matematicheskii Sbornik* 128.1 (1971), pp. 9–33.

INRIA Bordeaux–Sud-Ouest, 200 avenue de la Vieille Tour, 33405 Talence Cedex FRANCE
Email address: damien.robert@inria.fr
URL: http://www.normalesup.org/~robert/

Institut de Mathématiques de Bordeaux, 351 cours de la liberation, 33405 Talence cedex FRANCE

6 REDUCIBLE GLUING OF ABELIAN VARIETIES (2022-09)

These unfinished notes are available at http://www.normalesup.org/~robert/pro/publications/notes/2022-09-kani.pdf

The goal was to extend (part of) Kani's work in [Kan97] from elliptic curves to abelian varieties, beyond what is used nowadays in isogeny based cryptography. Like the preceding notes, I never got around to finishing them.

Still, the description of the maximal isotropic kernels in higher dimension might be useful.

Reducible gluing of abelian varieties

DAMIEN ROBERT

Abstract. This note extend Kani's work on reducible gluing of elliptic curves to abelian varieties.

1. Introduction

Kani's lemma [Kan97, § 2] has been a hot topic in isogeny based cryptography [CD22; MM22; Rob22a; Rob22b]. It is easy to extend it to abelian varieties, see [Rob22a, Lemma 3.4]. But Kani's work on reducible gluing of elliptic curves in [Kan97] goes further than just this lemma. The purpose of these notes is to cover the extension of Kani's work to higher dimensional abelian varieties. This is mostly a straightforward adaptation of Kani's proofs from elliptic curves to abelian varieties, with a few subtleties stemming from the fact that maximal isotropic kernels in abelian varieties are not always nicely described.

Kani's work in [Kan97, § 2] cover three related topics: how to combine an N_1-isogeny $f_1 : E_0 \to E_1$, and a N_2-isogeny $f_2 : E_0 \to E_2$ into an N_1+N_2-isogeny $F : E_0 \times E_0' \to E_1 \times E_2$, why they are all of this form, and describe the kernel of F. The applications mentioned above only really need the case where N_1 is prime to N_2 which simplify things. Nevertheless, the general case is interesting and Kani deals with it in details for elliptic curves. In Section 3 we show how his results extend to dimension g abelian varieties. But first we need to describe maximal isotropic subgroups in more details, this is done in Section 2.

Throughout these notes we only deal with separable isogenies. In particular, when looking at N-isogenies, we implicitly restrict to the case where N is prime to the characteristic p of the base field (or $p = 0$).

2. Maximal isotropic kernels

Let (A, λ_A) be a ppav.

Definition 2.1. A subgroup $K \subset A[N]$ is called isotropic (with respect to the Weil pairing $e_{A,N}$ on A) if $K \subset K^\perp$, ie if $e_{A,N}(P, Q) = 1$ for all $P, Q \in K$.

In the theory of bilinear form, such a subgroup K is usually called totally isotropic. An isotropic subgroup $H \subset G$ for a quadratic form q on G usually means that there is an isotropic element $x \neq 0 \in H$, ie such that $q(x) = 0$. However, since $e_{A,N}$ is alternating, every non trivial subgroup of $A[N]$ is isotropic in this sense.

Lemma 2.2. *Let $K \subset A[N]$ be a subgroup. The following are equivalent:*

1. *K is isotropic, and maximal among isotropic kernels (ie K is maximal isotropic);*
2. *$K = K^\perp$.*
3. *K is isotropic of cardinal N^g.*

Date: September 26, 2022.

Proof. 1 ⇒ 2 ⇒ 3 ⇒ 1, using that $\#K^\perp \#K = N^{2g}$ since $e_{A,N}$ is symplectic, which shows that in particular an isotropic group has cardinal $\#K \leq N^g$. □

Lemma 2.3. *Let K be a maximal isotropic subgroup and $\ell \mid \#K$. Then $K[\ell]$ is of dimension at least g (over $\mathbb{Z}/\ell\mathbb{Z}$), so contains a maximal isotropic subgroup K' for $A[\ell]$.*

Proof. By the symplectic CRT theorem, we may assume $N = \ell^e$. If $K[\ell]$ was of dimension $r < g$, we would have $\#K \leq (\ell^e)^r < (\ell^e)^g$, and K would not be maximal. So $K[\ell]$ is of rank $r \geq g$, hence we can always extract an isotropic subgroup of rank g by the structure theorem of symplectic vector spaces. (Be careful that K itself will not be isotropic for $A[\ell]$ if $r > g$). □

Corollary 2.4. *Every N-isogeny can be decomposed as product of ℓ_i-isogenies with $N = \prod \ell_i$.*

If K is a finite abelian group, we define its rank r has the minimal integer r such that there exists a surjection $\mathbb{Z}^r \to K$. This is also the number of elementary divisors $d_1 \mid d_2 \mid \dots d_r$ with $d_1 \neq 1$, giving the structure $K \simeq \prod \mathbb{Z}/d_i\mathbb{Z}$. This is also the maximum of the dimensions of the $K[p]$ as a $\mathbb{Z}/p\mathbb{Z}$ vector space over all p (dividing $\#K$). We call a "basis" of K a system of generators $(g_1, \dots, g_r)$ of cardinal r.

Lemma 2.5. *A maximal isotropic kernel $K \subset A[N]$ of rank g always has an isotropic complement K', meaning that $A[N] = K \oplus K'$ is a symplectic decomposition. In particular, if $(e_1, \dots, e_g)$ is a basis of K, it extends into a symplectic basis $(e_1, \dots, e_g, f_1, \dots f_g)$ of $A[N]$, and if $m \mid N$, $K[m]$ is maximal isotropic in $A[m]$.*

Proof. By the symplectic CRT theorem, we may reduce to the case $N = \ell^g$. Since K is of rank g and is of cardinal ℓ^g, it is homogeneous. It has a symplectic complement by [PSV10, Theorem 10.14]. □

Example 2.6. In $A[\ell]$ (with ℓ prime) an isotropic subgroup K is maximal iff it is of rank g (by the structure theorem of symplectic vector spaces).

Lemma 2.7. *If $K \subset A[\ell^e]$ is homogeneous (all its invariants are equal), it is either of rank g or of rank $2g$, In the latter case, $e = 2f$ and $K = A[\ell^f]$.*

If $K \subset A[N]$ is homogeneous or more generally if each ℓ-Sylow of K is homogeneous (this condition is equivalent to, if $d_1 \mid \dots \mid d_{2g}$ are the invariants of H where d_i is allowed to be 1, then each prime divisor ℓ of N divides at most one quotient d_{i+1}/d_i), then $N = N_1^2 N_2$ with $\gcd(N_1, N_2) = 1$, $K[N_1^2] = K[N_1] = A[N_1]$ and $K[N_2]$ maximal isotropic of rank g in $A[N_2]$.

Proof. By [PSV10, Theorem 10.14], K is standard (see below). Let $(e_1, \dots, e_g, f_1, \dots, f_g)$ a symplectic basis of $A[N]$ adapted to a standard decomposition $K = K_1 \oplus K_2$. Then since K is homogeneous, either K has for basis (say) $(e_1, \dots, e_k, f_{k+1}, \dots f_g)$ so it has rank g, or for basis $(\ell^f e_1, \dots, \ell^f e_g, \ell^f f_1, \dots \ell^f f_g)$ with $e = 2f$.

The general case comes from the symplectic CRT. □

It is often convenient to treat the case of maximal isotropic subgroups of the form $A[n]$ and those of rank g together. The following notion encompass these two cases:

Lemma 2.8. *Let $A[N] = A_1[N] \oplus A_2[N]$ be a symplectic decomposition. Let $K_1 \subset A_1[N]$ be any subgroup. Let $K_2 = K_1^\perp \cap A_2[N]$. Then $K = K_1 \oplus K_2$ is maximal isotropic. Conversely, if $K = K_1 \oplus K_2$ is maximal isotropic with $K_i \subset A_i[N]$, then $K_2 = K_1^? \cap A_2[N]$.*

Proof. K_2 is orthogonal to K_1 by definition, and orthogonal to itself because it lives in $A_2[N]$. Hence K is isotropic. We have $K_1^? = A_1[N] \oplus K_2$, so since $K_1^{?^?} = K_1$, $K_2^? = K_1 \oplus A_2[N]$. So $K^? = K_1^? \cap K_2^? = K$, hence K is maximal isotropic. The converse follows by the same calculation. □

Definition 2.9. A isotropic subgroup K is called standard if there is a symplectic decomposition $A[N] = A_1[N] \oplus A_2[N]$ such that $K = K_1 \oplus K_2$ where $K_i = K \cap A_i[N]$.

In particular, if $K = K_1 \oplus K_2$ is a standard isotropic subgroup, $K_2 \subset K_1^\perp \cap A_2[N]$, and by Lemma 2.8, K is maximal iff we have equality.

Example 2.10.
- a maximal isotropic kernel of rank g is standard by Lemma 2.5.
- a homogeneous maximal isotropic kernel is standard by Lemma 2.7.
- For an elliptic curve, a maximal isotropic subgroup $K \subset E[N]$ is always of the form $K = \langle P, Q\rangle$ where $P = me_1$, $Q = nf_1$ with (e_1, f_1) a symplectic basis of $E[N]$, $m \mid n$ and $N = mn$. In particular, K is standard.

  If $m = 1$, $K = \langle P\rangle$ is cyclic. If $m = n$, $K = E[n] \subset E[n^2]$. Since an isogeny $f : E \to E'$ of degree N is always an N-isogeny, $K = \operatorname{Ker} f$ is maximal isotropic, and f decomposes as $f = g \circ [m]$, where g has cyclic kernel, where m is as above for K.
- Let (e_1, e_2, f_1, f_2) be a symplectic basis of $A[\ell^2]$, A an abelian surface. Then $K = \langle e_1, e_2\rangle$ is maximal isotropic of rank $g = 2$.

  $K = \langle e_1, \ell e_2, \ell f_2\rangle$ is standard of rank 3. Notice that $K[\ell]$ is not isotropic in $A[\ell]$.
- In higher dimension, not every maximal isotropic kernel is standard [PSV10, Theorem 10.13].

The nice thing about standard maximal isotropic subgroups is that we can reduce to hyperbolic planes.

Lemma 2.11. *Let $A[N] = \oplus_{i=1}^g H_i$ be a symplectic decomposition of $A[N]$ into hyperbolic planes (ie a subgroup of rank 2 such that the symplectic forms stay non degenerate), $K_i \subset H_i$ an isotropic subgroup of H_i and $K = \oplus_{i=1}^g K_i$. Then K is standard isotropic, and is maximal iff each K_i is maximal in H_i.*

Conversely, if K is maximal standard isotropic, then there exists a symplectic decomposition $A[N] = \oplus_{i=1}^g H_i$ such that $K = \oplus_{i=1}^g K \cap H_i$.

Proof. By the symplectic CRT, we reduce to the case $N = \ell^e$. Each $K_i \subset H_i$ is standard in H_i by Example 2.10. Let (e_i, f_i) be a symplectic basis of H_i, these glue together to form a symplectic basis of $A[N]$, hence a symplectic decomposition of $A[N]$. This shows that $K = \oplus_{i=1}^g K_i$ is standard. Furthermore, $K^\perp \cap H_i = K_i^{\perp_{H_i}}$, so K is maximal in $A[N]$ iff each K_i is maximal in H_i.

Conversely, let $K = K_1 \times K_2$ be a decomposition of a maximal standard isotropic K induced by a symplectic decomposition of $A[N]$. Take $(e_1, \dots, e_g)$ a basis of $A_1[N]$ compatible with K, ie such that $K = \oplus_{i=1^g} K \cap \langle e_i\rangle$. This is possible because $A_1[N]$ is homogeneous. Then the dual basis $(f_1, \dots, f_g)$ of $A_2[N]$ with respect to $(e_1, \dots, e_g)$ is adapted to K_2 because $K_2 = K_1^\perp \cap A_2[N]$. Letting $H_i = (e_i, f_i)$, we get that $K = \oplus_{i=1}^g K \cap H_i$. □

3. Gluing abelian varieties

We generalize Kani's study of gluing of elliptic curves [Kan97, § 2] to the case of abelian varieties. As mentioned in the introduction, this is mostly a straightforward generalisation of his proofs.

3.1. **Gluing.** Let A, B be two abelian varieties of dimension g. A gluing $F : A \times B \to C$ is an isogeny from the product $A \times B$ to a dimension $2g$ abelian variety C. An uninteresting case is when F can be written as a diagonal isogeny $F = (f_1, f_2) : A \times B \to A' \times B'$ where $f_1 : A \to A'$ and $f_2 : B \to B'$ are two isogenies. More generally, if $\mathrm{Ker}\, F = H_A \times H_B$ then F is the composition of a diagonal isogeny followed by an automorphism. We call such a F a product gluing (because its kernel is a diagonal product).

We will look at the case when A, B are principally polarised, and F is an N-isogeny. Note that in the case of a product isogeny $F = (f_1, f_2)$, if f_1 is an N_1-isogeny and f_2 a N_2-isogeny, then F is a (N_1, N_2)-isogeny.

We will call F a *minimal gluing* if it does not factorize through such a *product gluing*. An equivalent condition is that $\mathrm{Ker}\, F \cap A \times 0 = \{0\}$ and $\mathrm{Ker}\, F \cap 0 \times B = \{0\}$. Let $g_1 = \dim A$, $g_2 = \dim B$, and F be a N-minimal gluing. Then the projections p_A and p_B are injective on $\mathrm{Ker}\, F$. Since $\mathrm{Ker}\, F$ is maximal isotropic in $(A \times B)[N]$, it is of cardinal $N^{g_1+g_2}$, so we get that $g_1 + g_2 \leq 2g_1$ and $g_1 + g_2 \leq 2g_2$, so $g_1 = g_2$. Henceforth, we let $g = g_1 = g_2$.

Lemma 3.1. *Let $\psi : A[N] \to B[N]$ be an anti-isometry with respect to the Weil pairing. Then $K = \{(P, \psi(P) \mid P \in A[N]\}$ is the kernel of a minimal N-gluing $F : A \to B \times C$. Conversely, the kernel $\mathrm{Ker}\, F$ of a minimal N-gluing is of this form. Furthermore, to check that F is minimal it suffices to check that $\mathrm{Ker}\, F \cap 0 \times B = 0$ or $\mathrm{Ker}\, F \cap A \times 0 = 0$.*

Proof. Let F be a minimal gluing and $K = \mathrm{Ker}\, F$. Since $K \cap 0 \times B = \{0\}$, the projection $p_A : A \times B \to A$ is injective on K. Since $K \subset (A \times B)[N]$ is maximal isotropic in the N-torsion, it is of cardinal N^{2g}, so the image of K is surjective in $A[N]$. Hence $p_A^{-1} : A[N] \to K$ is well defined, and composing with p_B we see that there is a well defined function ψ such that $K = \{(P, \psi(P)) \mid P \in A[N]\}$.

Since K is maximal isotropic, we get $e_{A\times B,N}((P_1, \psi(P_1)), (P_2, \psi(P_2))) = e_{A,N}(P_1, P_2)e_{B,N}(\psi(P_1), \psi(P_2)) = 1$, so $\psi : A[N] \to B[N]$ is an anti-isometry.

Conversely, the same computation shows that if ψ is an anti-isometry, $K = \{(P, \psi(P) \mid P \in A[N]\}$ is isotropic in $(A \times B)[N]$, hence is maximal isotropic since it is of cardinal N^{2g}. Furthermore, since ψ is an anti-isometry, it is injective (hence bijective), so $K \cap A \times 0 = 0$. This proves the last statement. □

Remark 3.2. Since $(1, -1)$ and $(-1, 1)$ are automorphisms of $A \times B$, the kernel $K' = \{(P, -\psi(P) \mid P \in A[N]\}$ also define a minimal N-gluing which is isomorphic to the one associated to K.

3.2. **Reducible gluing.** Now it can happen that in a minimal gluing $F : A \times B \to C$, C splits into a product even when F is not a product isogeny. We say that F is reducible.

When $g = 1$, in that case C splits as a product of elliptic curves, so $F = \begin{pmatrix} a & c \\ b & d \end{pmatrix}$ is automatically a matrix of n-isogenies (n depending on the component), because elliptic curves have their Neron-Severi group of rank 1 (ie is trivial). In dimension $g > 1$, C may not split into a product of two dimension g abelian varieties. And even if it does, the matrix $F = \begin{pmatrix} a & c \\ b & d \end{pmatrix}$ may not be given by individual n-isogenies if A or B has non trivial Neron-Severi group.

Definition 3.3. A (minimal) gluing $F : A \times B \to C$ is said to be reducible if $C \simeq A' \times B'$ with A', B' of dimension g, and $F = \begin{pmatrix} a & c \\ b & d \end{pmatrix}$ is given by a n_a-isogeny $a : A \to A'$, a n_b-isogeny $A \to B'$, a n_c-isogeny $c : B \to A'$ and a n_d-isogeny $d : B \to B'$. It is said to be non trivial reducible if F is not a product gluing.

Here by abuse of notation, we allow the case $n = 0$, where the notion of "0-isogeny" means that the morphism is 0 (so is not an actual isogeny).

Lemma 3.4. *Let F be a reducible N-gluing. Then $F = \begin{pmatrix} a & c \\ b & d \end{pmatrix}$, with $n_a = n_d$, $n_b = n_c$, $n_a + n_b = N$, $\tilde{c}a = -\tilde{d}b$.*

In particular, F is not diagonal iff $n_b = n_c \neq 0$.

Proof. The contragredient isogeny is given by $\tilde{F} = \begin{pmatrix} \tilde{a} & \tilde{b} \\ \tilde{c} & \tilde{d} \end{pmatrix}$, and the equation $\tilde{F}F = N$ gives $n_a + n_b = N$, $n_c + n_d = N$, $\tilde{a}c + \tilde{b}d = 0$, $\tilde{c}a + \tilde{d}b = 0$. By duality the last equation is already implied by the third one. The third equation also implies $n_a n_c = n_b n_d$, so $n_a = n_d$, $n_b = n_c$. □

3.3. **Isogeny diamonds.** Lemma 3.4 shows that the following notion is natural:

Definition 3.5. A (n_1, n_2)-isogeny diamond is a decomposition of a $n_1 n_2$-isogeny $f : A \to B$ between principally polarised abelian varieties into two different decompositions $f = f_1' \circ f_1 = f_2' \circ f_2$ where f_1 is a n_1-isogeny and f_2 is a n_2-isogeny. (Then f_1' will be a n_2-isogeny and f_2' a n_1-isogeny.) This decomposition is said to be minimal if $\operatorname{Ker} f_1 \cap \operatorname{Ker} f_2 = \{0\}$ (this is equivalent to the fact that f_1 and f_2 do not factorize through a common isogeny), and it is said to be orthogonal if n_1 is prime to n_2 (in which case it is automatically minimal).

$$\begin{array}{ccc} A & \xrightarrow{f_1} & A_1 \\ \downarrow{\scriptstyle f_2} & & \downarrow{\scriptstyle f_1'} \\ A_2 & \xrightarrow{f_2'} & B \end{array}$$

In [Kan97, § 2], Kani reserves the name isogeny diamond to what we call here a minimal isogeny diamond. We changed the term here slightly, because an isogeny diamond always induces a reducible gluing $F : A \times B \to A_1 \times B_2$, even if it is not minimal.

Remark 3.6.
If we have an isogeny diamond starting from A as above, taking duals where needed we also have an isogeny diamond starting from A_1, A_2 and B. If the isogeny diamond starting from A is minimal, we will see in the proof of Corollary 3.9 that the one from B is too, ie $\operatorname{Ker} \widetilde{f_1'} \cap \operatorname{Ker} \widetilde{f_2'} = 0$. However, the one from A_1 (or A_2) may not be minimal.

As a counterexample, take a symplectic decomposition $A[\ell] = K_1 \oplus K_2$, $f_1 : A \to A_1$ the quotient by K_1 and $f_2 : A \to A_2$ the quotient by K_2; f_1' the quotient of A_1 by $f_1(K_2)$ and f_2' the quotient of A_2 by $f_2(K_1)$. Then $f_1' : A_1 \to A$ is exactly the dual isogeny $\widetilde{f_1}$, so $\operatorname{Ker} f_1' \cap \operatorname{Ker} \widetilde{f_1} = \operatorname{Ker} f_1' \neq 0$.
An isogeny diamond is completely determined by (f_1, f_2, f). So it determines $H_1 = \operatorname{Ker} f_1$, $H_2 = \operatorname{Ker} f_2$ and $H = \operatorname{Ker} f$. In particular, H is maximal isotropic in $A[n_1 n_2]$, $H_1 \subset H$ maximal isotropic in $A[n_1]$, and $H_2 \subset H$ maximal isotropic in $A[n_2]$. Note that if $H_1 \cap H_2 = 0$ (ie the diamond is minimal), then $H = H_1 \oplus H_2$ since both members have the same cardinality.
When we have a commutative square as above, this square is a pushout iff $\operatorname{Ker} f = \operatorname{Ker} f_1 + \operatorname{Ker} f_2$ where $f = f_2' \circ f_2 = f_1' \circ f_1$. So a minimal isogeny diamond is a pushout square.

Conversely, if f is the pushout of a n_1-isogeny f_1 by a n_2-isogeny f_2, and $\gcd(n_1, n_2) = 1$, then f is a (orthogonal) isogeny diamond. But in general, the pushout f_1' of f_1 need not be an n_1-isogeny, in which case the pushout is not a diamond.

Lemma 3.7 (Kani). *Let $f = f_1' \circ f_1 = f_2' \circ f_2$ be a (n_1, n_2)-isogeny diamond as above. Then $F = \begin{pmatrix} f_1 & \widetilde{f_1'} \\ -f_2 & \widetilde{f_2'} \end{pmatrix}$ is a n-isogeny $A \times B \to A_1 \times A_2$ where $n = n_1 + n_2$. Furthermore, if f is minimal, $\ker F = \{(\tilde{f_1}, f_1' x), x \in A_1[n]\}$, and if f is an orthogonal isogeny diamond, then $\ker F = \{(n_1 x, fx), x \in A[n]\}$.*

Proof. For the product polarisations, the dual isogeny $\tilde{F}$ is given by $\tilde{F} = \begin{pmatrix} \widetilde{f_1} & \widetilde{f_2} \\ -f_1' & f_2' \end{pmatrix}$ and we directly check that $\tilde{F}F = (n_1 + n_2)\,\mathrm{Id}$. Furthermore, $\ker F$ is the image of $\tilde{F}$ on $A \times B[d]$, and if n_1 is prime to n_2 this is also the image of $\tilde{F}$ on $A[n] \times \{0\}$, so $\ker f = \{(\tilde{f_1} x, -f_1' x), x \in A[n]\} = \{(n_1 x, -fx), x \in A[n]\}$. □

Remark 3.8.

- One may of course permute f_1 and f_2, to get the same matrix F up to permutation of the coordinates. In terms of kernels, this amount to permuting H_1 and H_2 and replacing f by $-f$. It is not hard to prove that $\ker F$ is completely determined by (H_1, H_2, f), and that there is a bijection between the $\ker F$ for the isogeny diamonds, and the triplet (H_1, H_2, f) modulo the above equivalence: $(H_1, H_2, f) \equiv (H_2, H_1, -f)$. The exact same proof as in [Kan97, Theorem 2.3] (more precisely the first three paragraphs p. 9) hold.
- Since we have automorphisms $(-1,-1)$, $(-1,1)$ and $(1,-1)$ on $A \times B$, we can also use the matrix $F' = \begin{pmatrix} f_1 & -\widetilde{f_1'} \\ f_2 & \widetilde{f_2'} \end{pmatrix}$, whose kernel, in the case of an orthogonal isogeny diamond, is $\ker F' = \{(n_1 x, -fx), x \in A[n]\}$. In general, $\ker F' \neq \ker F$: there are two different reducible isogenies $A \times B \to A_1 \times A_2$.
- Note that F is not a product gluing, so in particular is a non trivial reducible gluing. Indeed if $\ker F$ was a digonal product $G_1 \times G_2$, we would have $G_1 \subset \ker f_1$, $G_2 \subset \ker \widetilde{f_2'}$. So $\#G_1 \leq n_1^g$, $\#G_2 \leq n_2^g$, but $\ker F = \#G_1 \#G_2 = n^{2g}$, which is a contradiction.

Corollary 3.9. *There is a bijection between triple (H_1, H_2, f) of isogeny diamonds modulo the equivalence defined above, and non diagonal maximal reducible kernels K of $A[n]$.*

This bijection induces an equivalence between minimal isogeny diamonds and minimal reducible gluing.

Proof. The first statement result from the combination of Lemmas 3.4 and 3.7 and Remark 3.8.

For the second statement, we need to prove that F is minimal iff $\ker f_1 \cap \ker f_2 = 0$. Note that $\ker F \cap A \times 0 = \ker f_1 \cap \ker f_2$, so one application is clear. For the converse, since $\ker F \cap 0 \times B = \ker \widetilde{f_1'} \cap \ker \widetilde{f_2'}$ we need to prove that if the diamond is minimal, $\ker \widetilde{f_1'} \cap \ker \widetilde{f_2'} = 0$, ie the diamond starting from B is also minimal. But this is a consequence of Lemma 3.1. □

References

[CD22] W. Castryck and T. Decru. *An efficient key recovery attack on SIDH (preliminary version)*. Cryptology ePrint Archive, Paper 2022/975. 2022. URL: https://eprint.iacr.org/2022/975.

[FC90] G. Faltings and C.-L. Chai. *Degeneration of abelian varieties*. Ergebnisse der Mathematik und ihrer Grenzgebiete (3) 22. Springer-Verlag, Berlin, 1990.

[Kan97] E. Kani. "The number of curves of genus two with elliptic differentials." In: *Journal für die reine und angewandte Mathematik* 485 (1997), pp. 93–122.

[MM22] L. Maino and C. Martindale. *An attack on SIDH with arbitrary starting curve*. Cryptology ePrint Archive, Paper 2022/1026. 2022. URL: https://eprint.iacr.org/2022/1026.

[Mil91] J. Milne. *Abelian varieties*. 1991. URL: http://www.jmilne.org/math/CourseNotes/av.html.

[MFK94] D. Mumford, J. Fogarty, and F. Kirwan. *Geometric invariant theory*. Vol. 34. Springer Science & Business Media, 1994.

[PSV10] A. Prasad, I. Shapiro, and M. Vemuri. "Locally compact abelian groups with symplectic self-duality". In: *Advances in Mathematics* 225.5 (2010), pp. 2429–2454.

[Rob22a] D. Robert. "Breaking SIDH in polynomial time". Aug. 2022. URL: http://www.normalesup.org/~robert/pro/publications/articles/breaking_sidh.pdf. eprint: 2022/1038.

[Rob22b] D. Robert. "Evaluating isogenies in polylogarithmic time". Aug. 2022. URL: http://www.normalesup.org/~robert/pro/publications/articles/polylog_isogenies.pdf. eprint: 2022/1068.

INRIA Bordeaux–Sud-Ouest, 200 avenue de la Vieille Tour, 33405 Talence Cedex FRANCE
Email address: damien.robert@inria.fr
URL: http://www.normalesup.org/~robert/

Institut de Mathématiques de Bordeaux, 351 cours de la liberation, 33405 Talence cedex FRANCE

7 A NOTE ON OPTIMISING 2^n-ISOGENIES IN HIGHER DIMENSION (2023-06)

These notes are available at http://www.normalesup.org/~robert/pro/publications/notes/2023-06-optimising_isogenies.pdf.

Most of the content (along with an efficient implementation [DMPR23b]!) is now available in [DMPR23a], but some of the result of these notes were not incorporated in that article due to space reason (beside they also deal with higher dimension than just two).

However, as explained in the introduction of these notes, most of their content are somewhat obsolete, due to the more efficient formula I discovered while working on [DMPR23a], so I recommend looking at [DMPR23a] for a much more streamlined presentation of the formulas.

A note on optimising 2^n-isogenies in higher dimension

DAMIEN ROBERT

Abstract. We give various optimisations for the computations of 2^n-isogenies in higher dimension. In particular, we explain how to compute 2^n-isogenies by pushing forward g points (a basis of the kernel) rather than 2^g points at each step. We detail the case of $g = 1$ and $g = 2$.

Contents

Date: November 13 2023.

1. Context

With the explosion of higher dimensional isogeny cryptography, a group of isogenies enthusiasts have gathered around a Zulip chat (this includes Pierrick Dartois, Sabrina Kunzweiler, Luciano Maino, Giacomo Pope, …).

The goal was to first start with dimension 2 2^n-isogenies, with a focus on improving Festa and the SIDH attacks, and to pave the way for the dimension 4 implementation of the verification in SQISignHD.

The git repository `https://github.com/GiacomoPope/Theta-Isogenies` contains code for 2^n-isogenies in dimension 2 via Richelot isogenies (+ splitting and gluing) in Mumford coordinates, via Kummer coordinates (with formula due to Sabrina Kunzweiler), and via theta coordinates (in level 2, hence also on the Kummer).

The goal was to optimize these three different models and compare them to each other. These notes were written in June 2023 to describe both 2^n-isogenies algorithms in theta coordinates and various optimisations (in any dimension) I had found compared to the algorithm described in [DLRW23, Appendix C.2].

Strangely, theta functions have somewhat a reputation of being hard to work with and slow (maybe because they can work in any dimension and any degree). Contrary to these expectations, isogeny formula are actually pretty fast in theta coordinates, and most notably for 2^n-isogenies in level 2: level 2 theta functions are precisely tailored so that the action by translation of the 2-torsion (more precisely the theta group) gives extremely fast isogeny images (see also the simplicity of the duplication formula Section 5). Notably a 2-isogeny image in dimension 1 is even faster in theta coordinates than in Montgomery coordinates (see Section 15).

Of course, operation count never beat actual profiling, which was the goal of our Sage implementation (further comparison between the different dimension 2 models are out of scope of these notes, but theta functions are indeed very fast! The current implementation gives a factor 9× for the computation of a 2^{602}-isogeny chain, and images 17× faster, compared to Richelot isogenies.)

Although image computation is naturally very fast in theta coordinates, the codomain computation was originally a lot more involved. The original algorithm of [DLRW23, Appendix C.2]; involved a normalisation process involving 2^g points of 8-torsion.

The original goal of these notes involved a faster normalisation process involving only $1 + g$ points of 8-torsion, with some further optimisations like inlining what was needed for the tripling formula used in the normalisation process. This is the version which was first implemented in the git repository, by the people mentioned above.

Since then, I have found (in July 25) newer formula that completely bypass the normalisation process, see Section 17. These formula are both much simpler to implement[1] and a lot faster, they essentially boil down to g images computations.

These makes most of these notes obsolete, notably Sections 7, 10, 11, 14 and 16. These notes are still in their state of June 2023, except for this section and the newer Section 17, written in August 2023.

The obsolete normalisation process described in these notes for 2-isogenies might still have an interest to better explain the similar normalisation process using for higher degree ℓ-isogenies. Indeed, for $\ell > 2$ (and with theta functions of level 2) a normalisation process is needed both for codomain computations but also for images computations.

[1] I recommend looking at the git history to compare the old formula with the newer ones

One might wonder why this normalisation process is no longer needed for $\ell = 2$ but still needed for $\ell > 2$. The answer is that with theta coordinates in level 2, the points of 2 torsion are already normalised with respect to each others, hence the normalisation process was redundant. To have a similar process for $\ell = 3$ (say), we would need to work with theta functions of level 3 or 6. The normalisation process of the points of 3-torsion is essentially a way to work in level 2 almost as if we were in level 6.

Apart from the results of Section 17, we give several formula in the dimension 1 case that might be of interest in Section 15.

A word of warning: these are notes, not a research paper, and there are probably still a lot of remaining typos in the formula. When in doubt look at the code itself, it should be correct!

Update November 2023: we now have a paper [DMPR23a] detailing the formulas for a dimension two 2^n-isogeny in the theta model. The code is also available [DMPR23b]. Pierrick Dartois is working on a follow up paper for the adapation to dimension 4. We strongly recommend reading this article rather than these notes, which as mentioned organically grew as we went along and implemented the algorithm, so are not very readable!

2. Introduction

Computing isogenies in higher dimension has received considerable interest recently: breaking SIDH, SQISignHD, Festa [CD23; MMPPW23; Rob23a; DLRW23; BMP23]. Although algorithms in any dimensions are described in [LR12; CR15; LR15b; LR22a] in a theta model of even level n, for simplicity only the case of an ℓ-isogeny with ℓ prime to n is considered in these articles. For cryptographic applications, the most interesting case is when $\ell = 2^u$ and $n = 2$, which does not satisfy these conditions. The general case of ℓ non prime to n case is briefly treated in [Rob10, Proposition 6.3.5; Rob21, Remarks 2.10.3, 2.10.7 and 2.10.14]. A particular difficulty when ℓ is even is that we need a symplectic basis of the ℓn-torsion which is compatible with the symmetric level n theta structure, a condition for compatibility, due to David Lubicz, is described in [Rob21, Remark 2.10.7]. In an upcoming article with David Lubicz, we will treat this general case in more detail, along with algorithms to raise and descend the level (which are strongly linked to isogeny algorithms).

The purpose of these notes by contrary is to look only at speeding up the formula for the computation of the specific case of 2^n-isogenies in level 2. As usual, this rely on splitting the isogeny $\phi : A \to B$ into a product of n 2-isogenies ϕ_i, and push forward points by the ϕ_i, so we reduce to 2-isogenies. Building on [Rob10, § 6, § 7; Rob21, § 2, § 4], an algorithm to do so was presented in [DLRW23, Appendix C.2]; we will reuse the general notations of this article. For our cryptographic application, our isogeny $\phi : A \to B$ has for domain $A = \prod E_i$ a product of elliptic curves. This also simplifies various steps of the algorithm, notably the initialisation of the algorithm. Also, the compatibility conditions alluded to above is easy to verify in dimension 1 (see Lemma 8.3), and can be propagated through the product theta structure. This allows to essentially bypass it entirely in what follows.

Given $K = \langle T_1, \dots, T_g \rangle$ an isotropic kernel of A, the standard method to split the isogeny into 2-isogenies is to first compute a basis of $K[2]$ via doubling formula, compute the isogeny $\phi_1 : A \to A_1 = A/K[2]$, push the points T_i via ϕ_1, compute a basis of $f(K)[2]$ via a combination of doubling and pushing points via ϕ_1 (the optimal strategy depends on the relative cost of doubling and pushing points, given these costs an algorithm is described in [DJP14, § 4.2.2]).

We assume that we are given a theta null point of level 2 on A and that K is compatible with this theta null point (see Section 9). Given the theta null point of the isogeneous abelian

variety $A_1 = A/K[2]$, the theta model has particularly nice formula to compute the image by a point (see Section 5); this cost $2^g S + (2^g - 1)M$ assuming the theta constants of A_1 are normalised so that $\theta_0^{A_1}(0) = 1$ and the inverse $1/\theta_i^{A_1}(0)$ have been computed. Also it is possible to recover the squares $\theta_i^{A_1}(0)^2$ in only $2^g S$. Given the simplicity of these formula, doubling and differential addition on A are computed by going through the 2-isogeny to A_1 (see Section 6). In particular, doubling essentially cost 2 isogeny evaluations. Furthermore, for the arithmetic on A, the squares $\theta_i^{A_1{}^2}(0)$ are enough. However, for computing a 2^n-isogeny as a chain of 2-isogenies, we actually need the correct square roots $\theta_i^{A_1}(0)$.

A big part of this article is to optimize the formula to obtain these correct square roots. Let us explain the main idea, using $g = 2$ as an exemple. We have the theta null point $(a : b : c : d)$, and we can easily compute the squares of the dual coordinates of the isogenous theta null point $(A : B : C : D)$ via $(A^2 : B^2 : C^2 : D^2) = H \circ S(a : b : c : d) = H(a^2 : b^2 : c^2 : d^2)$ where H is the Hadamard transform and S is the squaring operation, ie $A^2 = a^2+b^2+c^2+d^2, B^2 = a^2-b^2+c^2-d^2, C^2 = a^2+b^2-c^2-d^2, D^2 = a^2-b^2-c^2+d^2$. If we have suitable points of 4-torsion T_1, T_2, then $(AB : CD : AB : CD) = H \circ S(T_1)$, $(AC : BD : AC : BD) = H \circ S(T_2)$, $(AD : BC : BC : AD) = H \circ S(T_1 + T_2)$. This is not enough to recover (A, B, C, D) because we are dealing with projective coordinates. What we really need is to recover (AB, CD, AB, CD) exactly. This can be done via a normalisation procedure. In other words, computing the isogenous theta constant can be done from the coordinates on some points of 4-torsion and a suitable normalisation procedure. This is not specific to the case $\ell = 2$, as mentioned above the general case of ℓ prime to n is [LR12; CR15; LR15b; LR22a] and the relatively straightforward adaptation (assuming that we are given *compatible* points of 4-torsion) to all cases is in [Rob10, Proposition 6.3.5; Rob21, Remarks 2.10.3, 2.10.7 and 2.10.14], and a more detailed algorithm for $\ell = n = 2$ given in [DLRW23, Appendix C.2].

The normalisation procedure exploit the (algebraic) Riemann relations, as constructed by Mumford in [Mum66] (see also [Rob10, Théorème 4.4.6]). These Riemann relations follow from the duplication formula, whose algebraic version was proved by Mumford in [Mum66] (see also [Rob10, Théorème 4.4.3]). The duplication formula is particularly well suited for the algorithmic of 2-isogenies, and in these notes we will exploit it as much as possible in order to speed up the generic algorithm working for any ℓ.

We describe two optimisations compared to [DLRW23, Appendix C.2].

(1) To compute the correct square roots, the equations in [DLRW23, § C.2] (derived from the duplication formula, see [Rob10, § 4.3]) require 2^g 4-torsion points in $K[4]$ (suitably normalised from our 8-torsion points), including the theta null point. This means that when we decompose ϕ, we need to push along 2^g-points at each step (or more precisely compute the isogenous theta null point and then push $2^g - 1$ points). In this note we give a new algorithm that only require the g generators of $K[4]$ along with the theta null point. So once we have computed the isogenous theta null point, we only require to push g points for the next step. The total gain is almost $(2^g - 1)/g$: while there is no difference for $g = 1$, for $g = 2$ we go from needing to keep track of 3 (non null) points to only 2, and for $g = 4$ from 15 points to only 4.

(2) Still to compute the correct square roots, a normalisation procedure is applied in [DLRW23, § C.2] (described in more details in [Rob10, § 6.3, § 7.4]) to some points of 8-torsion in $K[8]$. This normalisation procedure amount to choosing some "correct" choice of affine lift; and it is repeated for each 2-isogeny ϕ_i: for ϕ_2 we will normalise points of 8-torsion in $\phi_1(K)[8]$ and so on. Instead, we propose to

normalize once and for all the g generators T_i of K. Essentially this amount, once an affine lift of the theta null point of A is chosen, to choose consistent lifts of the T_i with respect to this lift. This means that from now on, all our algorithm have to work on affine lifts. Luckily all our algorithms are derived from the Riemann relations and duplication formula which naturally preserve this compatibility, so the compatibility is already "baked-in". Note that if compatible lifts $\widetilde{0_A}, \widetilde{T_i}$ are chosen, then the lifts $\lambda \star \widetilde{0_A}, \star\widetilde{T_i}$ are still compatible as long as λ does not depend on i. This allows for some optimisation: for instance it is harmless to choose a different normalisation of the theta null point of A_1, as long as this different normalisation is taken into account when pushing points.

The main advantage of normalising generators of K at the start is that when A is a product of elliptic curves, the normalisation procedure can be done in dimension 1.

Points on an abelian variety in the theta model are represented by projective points, but as explained above, at various points in the isogeny computations we need to work with affine lifts. All our algorithms will be on affine lifts by default; the projective version follows trivially.

3. The two torsion on a level 2 theta structure

Let $(A, \mathcal{L}, \Theta_A)$ be a principally polarised abelian variety with a symmetric theta structure of level 2. Let $0_A = (a_i)_{i\in Z(\overline{2})}$ be the theta null point.

The translation map by points of two torsion is defined as follows: the two torsion is isomorphic to $Z(\overline{2}) \times \hat{Z}(\overline{2})$, with $Z(\overline{2}) = \mathbb{Z}^g/2\mathbb{Z}^g$, and $\hat{Z}(\overline{2})$ its dual. If $P = (x_i)$ is an affine lift of a point on A, and T the two torsion point corresponding to (j, χ), $P + T = (\chi(i)x_{i+j})$.

Applying this to the theta null point, we recover the theta coordinates of the points of 2-torsion. Fixing the canonical basis $(e_1, \dots, e_g)$ of $Z(\overline{2})$, and letting f_i be the dual character of e_i, via our identification above the basis (e_i, f_i) is the canonical symplectic basis of the 2-torsion induced by theta theta structure.

Example 3.1. When $g = 1$, the theta null point is given by $(a, b) = (a_0, a_1)$. We have $e_1 = (b, a), f_1 = (-a, b)$. Dimension 1 is special in that we also have an explicit description of points of 4-torsion: $e_1' = (1 : 1)$ is the canonical point of 4-torsion above e_1 (the other one is $e_1' + f_1 = (-1 : 1)$), and $f_1' = (1 : 0)$ the canonical point of 4-torsion above f_1 (the other one is $f_1' + e_1 = (0 : 1)$).

Example 3.2. When $g = 2$, the theta null point is given by $(a_{00}, a_{01}, a_{10}, a_{11})$. We have $e_1 = (a_{01}, a_{00}, a_{11}, a_{10}), e_2 = (a_{10}, a_{00}, a_{11}, a_{01})$ and $e_1 + e_2 = (a_{11}, a_{10}, a_{01}, a_{00})$. We have $f_1 = (a_{00}, -a_{01}, a_{10}, -a_{11}), f_2 = (a_{00}, a_{01}, -a_{10}, -a_{11})$ and $f_1 + f_2 = (a_{00}, -a_{01}, -a_{10}, a_{11})$.

4. The Hadamard transform

Let H be the Hadamard matrix, given by $H_{i,\chi} = \chi(i)$. The action of H corresponds to the action of the modular matrix $\mathcal{S} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$; in particular this transpose the e_i with the f_i.

Starting with the theta coordinate θ_i, the coordinates θ'_χ resulting from the action of H are called the dual theta coordinates.

Example 4.1. When $g = 1, H(x, z) = (x + z, x - z)$.

One needs to be careful that $H \circ H = 2^g$ Id. This is not a problem in projective coordinate, but in affine coordinate we need to use $H^{-1} = H/2^g$.

5. The duplication formula

Let $K = \langle f_1, \dots, f_g \rangle$ and $f : A \to B = A/K$ the quotient. There are several ways to descend $\mathcal{L}^2$ to a principal polarisation $\mathcal{M}$ on B, but they all give the same totally symmetric line bundle $\mathcal{M}^2$ which is also the descent of $\mathcal{L}^4$ by the unique symmetric lift of K in the theta group $G(\mathcal{L}^4)$ which extends to a totally isotropic subgroup. Fix a compatible symmetric theta structure of level 2 on B.

Define the operation $\star$ by $(x_i) \star (y_i) = (x_i y_i)$. As a special case of the duplication formula, we have:

$$\theta^A(P+Q) \star \theta^A(P-Q) = H(\theta'^{,B}(f(P)) \star \theta'^{,B}(f(Q))) \tag{1}$$

$$H(\theta^A(\tilde{f}(R)) \star \theta^A(\tilde{f}(S))) = \theta'^{,B}(R+S) \star \theta'^{,B}(R-S) \tag{2}$$

This is the key for all our formula. First, using $Q = 0$, we can compute the image of a point by f via the operations

$$P = (\theta_i(P)) \xrightarrow{S} (\theta_i^2(P)) \xrightarrow{H} (\theta'^{,B}_\chi(f(P))\theta'^{,B}_\chi(0)) \xrightarrow{C_1} \theta'^{B}_\chi(f(P)) \xrightarrow{H} \theta_i^B(f(P)) \tag{3}$$

(Note: here and in what follows, we are probably off by some factor 2^g here; as long as this factor is uniform across all points this is ok. Also here S is the squaring map, not the modular matrix $\mathcal{S}$ from before).

Here the constants C_1 are given by $(1/\theta'^{,B}_\chi(0))$, the inverse of the dual theta coordinates of the theta null point on B. It is easy to compute their squares: $\theta'^{,B}_\chi(0)^2 = H(\theta_i^A(0)^2)$.

The remainder of this paper is devoted to compute the correct square roots of these squares. Note that compared to [DLRW23, Appendix C.2] we consider the isogeny with kernel $\langle f_i \rangle$ instead of the one with kernel $\langle e_i \rangle$. (We made a different choice in [DLRW23, Appendix C.2] because we used the analytic formalism, where the above choice was slightly more natural. In the algebraic formalism, it is slightly more natural to use our choice here. This does not matter much, because via H we can go from the coordinates to the dual coordinates. That's why our formula differ from [DLRW23, Appendix C.2] by the conjugation by H.)

6. Differential additions

We can also compute differential additions on A this way. First we compute $f(P)$ and $f(Q)$ using the differential addition formula as above, ie using them on the couple $(P, 0)$ and $(Q, 0)$. Then we use them again (in the other direction) to recover $\theta_i^A(P+Q)\theta_i^A(P-Q)$ from $(f(P), f(Q))$.

We actually don't need the $\theta'^{,B}_\chi(0)$, only their square, the trick is to start with P and only do the map $H \circ S$ to get $(\theta'^{,B}_\chi(f(P))\theta'^{,B}_\chi(0))$, same with Q. Then we apply the $\star$ operation on these coordinates to get $(\theta'^{,B}_\chi(f(P))(\theta'^{,B}_\chi(f(Q))\theta'^{,B}_\chi(0)^2)$, and now we can use C_1^2 to clear the extra factor $(\theta'^{,B}_\chi(0)^2)$. Using $Q = P$ we get the doubling map.

From the doubling and differential addition map, we can use the Montgomery ladder to compute the scalar multiplication on affine coordinates.

Example 6.1. When $g = 1$, $P = (x : z)$, we compute $f(P) = (r : s)$ by doing $(x : z) \xrightarrow{S} (x^2 : z^2) \xrightarrow{H} (x^2 + z^2 : x^2 - z^2) \xrightarrow{C} ((x^2 + z^2)/A : (x^2 - z^2)/B)$. We can compute $f(Q) = (u : v)$ in a similar way. Then $H((P+Q) \star (P-Q)) = H(f(P)) \star H(f(Q))$, so we compute $((r+s)(u+v) : (r-s)(u-v))$ (at this step we only need $(A^2 : B^2)$ which can be computed via $(A^2 : B^2) = (a^2 + b^2 : a^2 - b^2)$) and apply H to it to recover $(x(P+Q)x(P-Q) : z(P+Q)z(P-Q))$.

We also recover exactly Gaudry's addition formula for $g = 2$.

The doubling and differential addition algorithm assume that we are in the generic case and that none of the coordinates are zero. The general case is treated in [LR16], another solution is to apply any linear change of variable coming from the symplectic modular action (e.g., the action of H), we refer to Appendix B for the algebraic description of this action.

7. Normalising points

In [DLRW23, Appendix C] we explain how to use points of 4-torsion to compute the correct choice of $\theta'^{,B}_{\chi}(0)$. A key step is a normalisation procedure, and we actually need the points of 8-torsion to correctly normalize our points of 4-torsion (see also [Rob10; Rob21; LR22a]).

Lemma 7.1. *Let $\widetilde{P}$ be an affine point. Then $m(\lambda \star \widetilde{P}) = \lambda^{m^2} \star (m\widetilde{P})$.*

Let T be a 2-torsion point in our kernel $K = \langle f_1, \dots, f_g \rangle$. Let T'' be a point of $4m$-torsion above T, ie $T = mT''$. Write $2m = 2m_1 + 2$. We have $(m_1 + 2)T'' = -(m_1 T'') + T$.

Definition 7.2. Fix an affine lift $\widetilde{T''}$ of T''. We say that $\widetilde{T''}$ is normalised if $(m_1 + 2)\widetilde{T''} = -(m_1\widetilde{T''}) + \widetilde{T}$, where the action of translation by $\widetilde{T}$ is the affine one described in Section 3.

Lemma 7.3. *Fix an arbitrary affine lift $\widetilde{T''}$. By computing $(m_1 + 2)\widetilde{T''}$ and $(m_1)\widetilde{T''}$, we can find an equation $\lambda^{4m} = C$ such that for any solution λ, $\lambda \star \widetilde{T''}$ is normalised.*

Proof. Follows from Lemma 7.1. □

Example 7.4. Assume T'' is a point of 2^n-torsion above T. Applying the normalisation procedure of Lemma 7.3 to an arbitrary lift $\widetilde{T''}$, we get that $\lambda \star \widetilde{T''}$ is normalised for λ satisfying some equation $\lambda^{2^n} = C$. Then $2^{n-2}(\lambda \star \widetilde{T''}) = \lambda^{2^{2n-4}} \star (2^{n-2}\widetilde{T''})$ by Lemma 7.1.

It follows that if $n \geq 4$, the point T'' uniquely determines an affine lift $\widetilde{T'}$ of the point of 4-torsion $T' = 2^{n-2}T''$ above T. If $n = 3$, $\widetilde{T'}$ is uniquely determined up to a sign. Since the isogeny formula starts by the square operator S, this sign won't matter, so $n = 3$ is enough to normalize our points of 4-torsion.

Example 7.5. Let us start with $T' = (1 : 0)$, the canonical point of 4-torsion above $T = (a : -b)$ in dimension 1. We take the lift $\widetilde{T'} = (1, 0)$. We compute $2\widetilde{T'} = (\frac{a}{A^2B^2}, \frac{-b}{A^2B^2})$. The correct normalisation is thus $\lambda \star \widetilde{T'} = (\lambda, 0)$ with $\lambda^4 = A^2B^2$.

8. The choice of the theta constant for a 2-isogeny

When we apply Equation (3) to compute the image of a point by our isogeny, we have fixed the kernel of our 2-isogeny f to be $K = \langle f_1, \dots, f_g \rangle$.

If we start with another kernel, we need to apply an automorphism of the theta structure so that K corresponds to the $\langle f_1, \dots, f_g \rangle$ of the new theta null point; for instance if $K = \langle e_1, \dots, e_g \rangle$ the automorphism is the one given by the Hadamard transform.

A general procedure is as follow. First recall that the theta null point is induced by a symplectic basis of the 4-torsion. Fix such a basis $(e'_1, \dots, e'_g, f'_1, \dots, f'_g)$ inducing our theta null point. Let $(T_1, \dots, T_g)$ be a basis of K, choose any isotropic basis $(T'_1, \dots, T'_g)$ of 4-torsion point above the T_i, and complete the T'_i via a symplectic basis $(S'_1, \dots, S'_g, T'_1, \dots, T'_g)$. Compute the symplectic base change of matrix from $(e'_1, \dots, e'_g, f'_1, \dots, f'_g)$ to $(S'_1, \dots, S'_g, T'_1, \dots, T'_g)$, one can use the Weil pairing (an algorithm in theta coordinate is given in [LR10; LR15a; Rob21]) to compute this matrix M. Then apply the theta transformation formula for M.

It remains to explain how to fix $(e'_1, \dots, f'_g)$. As explained in the introduction, the general case will be treated in an upcoming article with David Lubicz. For our applications, we will use that A is a product of elliptic curve, so we only need to deal with $g = 1$ and use the fact that the product theta structure behaves as expected with respect to the symplectic basis:

Lemma 8.1. *If 0_A is induced by a basis $(e'_1, \dots, e'_{g_1}, f'_1, \dots, f'_{g_1})$ on A and 0_B is induced by a basis $(m'_1, \dots, m'_{g_2}, n'_1, \dots, n'_{g_2})$ on B, then the theta null point $(\theta_i^A(0)\theta_j^B(0))$ of the product theta structure is induced by the symplectic basis $((e'_i, 0), \dots (0, m'_j), \dots (f'_i, 0), \dots (0, n'_j))$ on $A \times B$.*

We are thus reduced to give a compatible symplectic basis of the four torsion in dimension 1. This case is easy because on the theta model of level 2 we always have the full 4-torsion (on the Kummer) when $g = 1$ (this is specific to the dimension 1 case).

Lemma 8.2. *On a theta model in dimension 1, a symplectic basis of $E[4]$ is given by $T'_1 = (1 : 1), T'_2 = (1 : 0)$.*

Proof. Let $(a : b)$ be the theta null point. Above $T_1 = (b : a)$ we have two points of 4-torsion (on the Kummer): $(1 : 1)$ and $(-1 : 1)$. Only one of the two is compatible with the theta structure. To determine which we use an idea due to David Lubicz: from a compatible four torsion point $T' = (u : v)$ we can compute a level 4 isogenous theta null point $(a, \lambda u, b, \lambda v)$, for λ an appropriate normalisation factor (see [Rob10; Rob21]). This level 4 theta null point has to be symmetric, which implies $\lambda u = \lambda v$. So we have $T'_1 = (1 : 1)$.

Above $T_2 = (-a : b)$ we have two points of 4-torsion: $(1 : 0)$ and $(0 : 1)$. The Hadamard transform of the first one is $(1 : 1)$ while for the second one we get $(1 : -1)$, so the correct compatible point is $T'_2 = (1 : 0)$. □

We can use the lemma above to convert a basis of 4-torsion (T'_1, T'_2) in a Montgomery model to a theta null point induced by this basis.

Lemma 8.3. *Let E be a Montgomery curve. Let $T'_1 = (1 : 1)$ be the canonical point of 4-torsion on the Kummer line in the Montgomery model. Let $T'_2 = (r : s)$ be another point of 4-torsion (with $2T'_2 \neq 2T'_1$). Then the theta null point associated to the basis (T'_1, T'_2) is $(a : b) = H(T'_2) = (r + s, r - s)$.*

Proof. This follows by looking at the ramification of the Kummer map $E \to E/\pm 1$ on our different models, see [BRS23]. □

We can use the above lemma on an arbitrary curve E with two explicit points T'_1, T'_2 of 4-torsion (with $2T'_1 \neq 2T'_2$) by first converting E to Montgomery form with T'_1 sent to $(1 : 1)$ and T_1 to $(0 : 1)$. This map is given by the homography $x \mapsto (x - x_0)/\beta$ with $x_0 = x(2T'_1)$ and $\beta = x(T'_1) - x_0$. See Appendix A for more details on converting to theta coordinates.

Example 8.4 (Dimension 2)**.** If we have two elliptic curves E_1, E_2 given by the theta constants $(a_1 : b_1), (a_2 : b_2)$, then the theta constant on $E_1 \times E_2$ is $(a_1a_2 : a_1b_2 : a_2b_1 : b_1b_2)$. And if $P_1 = (x_1 : z_1) \in E_1, P_2 = (x_2 : z_2) \in E_2$, $(P_1, P_2) = (x_1x_2 : x_1z_2 : x_2z_1 : z_1z_2) \in E_1 \times E_2$.

Remark 8.5. We briefly explain how the general case would work.

Let T'_i be a point of 4-torsion above $T_i \in K_2$. Then we have $T'_i + T_i = -T'_i$, hence in level 2, since we are on the Kummer, $(\theta_j(T'_i + T_i)) = (\theta_j(T'_i))$ in projective coordinates. From the action of T_i described in Section 3, we get that either $\theta_j(T'_i) = 0$ for all j such that $\chi_i(j) = 1$ or $\theta_j(T'_i) = 0$ for all j such that $\chi_i(j) = 0$. The compatibility conditions holds if we are in the first case for all i (this follows by a counting argument).

For instance, when $g = 2$, we should have $T'_1 = (x : 0 : z : 0)$ and $T'_2 = (u : v : 0 : 0)$. If $T'_1 = (0 : x : 0 : z)$ or $T'_2 = (0 : 0 : u : v)$ then these points are not compatible. This criteria can be used to check if our symplectic base change was correct.

Another difficulty in the general case, is that the 4-torsion is not immediately accessible (unlike the case for a product of elliptic curve). So we would first need to compute a symplectic basis (e'_i, f'_i) of the 4-torsion above the one (e_i, f_i) of the 2 torsion compatible with our current theta null point to apply the above strategy. In dimension 2 a method is described in Section 16.6, but this involves square roots.

We suggest the following alternative strategy: work only with the 2-torsion, and compute the symplectic base change matrix $M \in \mathrm{Sp}_{2g}(\mathbb{Z}/2\mathbb{Z})$. It is easy to express our 2-torsion points T_i in terms of the e_i, f_i: essentially the Weil pairing is trivial to compute in level 2 (namely check the translation which match the coordinates up to a sign, and then look at the signs). Lift M to an arbitrary matrix $\widetilde{M} \in \mathrm{Sp}_{2g}(\mathbb{Z}/4\mathbb{Z})$. While the points $\widetilde{M}.T_i$ will be correct by construction, the points $\widetilde{M}.T'_i$ probably won't be correct: the zeros will not be in the right position. But we can correct this via the action of $\Gamma(2,4)/\mathrm{Sp}_{2g}(\mathbb{Z}/2\mathbb{Z})$, essentially this acts like the translation of the 2-torsion so the correction is easy.

9. The choice of theta constants for a 2^n-isogeny

Let K be an isotropic 2^n-kernel of rank g on A. We want to first compute the quotient $f : A \to B = A/K[2]$, and then compute $f(K)$ in B, and recurse our formula.

First, our kernel $K[2]$ has to be compatible with our chosen theta null point on A, as explained in Section 8. Then as explained in Section 5, it is easy to recover the squares of the dual theta coordinates of B.

While we can prove that any choice of square roots of these coordinates correspond to an honest (dual) theta null point on B when $g \leq 2$, this is no longer the case in higher dimension, once we have fixed some square roots the other ones have to satisfy some compatibility condition.

Most importantly, our choice of theta constant on B determines the next 2-isogeny. But we want to compute the isogeny with kernel K, so our next isogeny has to be $f(K[4])$! So we do not want arbitrary (compatible) square roots anyway, but the ones which correspond to $f(K[4])$.

There is one remaining subtlety. Our theta constant on A determines a bit more than the symplectic basis of 2-torsion (hence the kernel of the first 2-isogeny). It is enough to fix a symplectic basis of the 4-torsion (and several such basis will determine the same theta null point). This means that we also require some compatibility between our kernel K and our theta null point on A: let $(f'_1, \dots, f'_g)$ be a basis of $K[4]$ with $f_i = 2f'_i$. Our first compatibility condition was that f_i is the canonical point of 2-torsion induced by our theta structure as described in Section 3, ie $K[2]$ and our theta null point are compatible. We require furthermore that our theta null point is induced by some symplectic basis $(e'_1, \dots, e'_g, f'_1, \dots, f'_g)$, in which case we say that $K[4]$ and our theta null point are compatible.

But now for our choice of sign for the theta null point of B, we want this theta null point to be compatible with $f(K)[4]$. Since $f(K)[4] = f(K[8])$, we will also need to use the points of 8-torsion in the kernel to fix our sign choice.

Given $K[8]$, it is possible to use these points of 8-torsion to normalize the points of 4-torsion up to factors $\lambda^2 = C$ as explained in Section 7. Since the choice of signs for the dual theta null point of B depends only on the square of the theta coordinates of these points

of 4-torsion (see [DLRW23, Equation (6) and (8)] or the duplication formula in [Rob10, Théorème 4.4.3]) this is enough to completely determine the theta null point of B.

A problem remains at the last 2 steps of the isogeny chain, when we only have access to 4-torsion points (resp. 2-torsion points) in K. It is possible to show that when the 2-torsion on B is not fixed by $K[4]$, there are $g(g+1)/2$ possible choice of signs for the dual theta null point of B. This follows by looking at the possible automorphisms of the theta structure as in [Rob10, § 6.3]. If we have $K[4]$ but not $K[8]$, we only have g possible choice of signs: the isotropic part $f(K[4])$ of the 2-torsion on B is fixed but we can change the symmetric lifts above them. These sign can be determined as follow: take $T'_1, \dots, T'_g$ a basis of $K[4]$ and normalize these points, we obtain equations $\lambda_i^4 = C_i$. The points $T'_i + T'_j$ can then be normalised up to some equation $\lambda_{ij}^2 = C_{ij}$, and from these all other points can be computed from extended Riemann relations, notably the three way additions. Since the theta null point on B only depend on the squares of the theta coordinates of the normalised points of 4-torsion, we obtain our g-choice of sign corresponding to the choices of $\lambda_i^2 = \pm\sqrt{C_i}$.

There are many reason to want more control on these last two steps. Typically for cryptographic applications, the codomain B of ϕ is also a product of elliptic curves, and we want to map back to these curves. This is easy to do if the theta null point $\theta_i(0_B)$ comes from a product theta structure, but there is no reason for this to be the case. One would then need to take an automorphism of the theta structure which brings it to a product theta structure. Also it is often the case that the isogeny $\phi : A \to B$ is split as an isogeny $\phi_1 : A \to C$ and a dual isogeny $\widetilde{\phi}_2 : B \to C$. One then need to glue together the theta null point computed on C from ϕ_1 and $\widetilde{\phi}_2$, they have no reason to be induced by the same theta structure, hence be the same. Again they will differ by some automorphism of the theta structure. As carefully explained in [DLRW23, § C.1], by keeping track of a bit more torsion it is possible to compute in advance the correct automorphism of the theta structure that we need in these computation. This means that our algorithm will start with K' an isotropic kernel of rank g of $A[2^{n+2}]$, and we compute the quotient $B = A/K$ where $K = K'[2^n]$ and the theta null point on B is the one induced by the theta null point on A along with our choice of K' (half the information given by the theta null point on A is killed by our isogeny ϕ, and K' allows precisely to uniquely recover this missing information).

10. Normalising the points for a 2^n-isogeny

From now on we suppose that we have K' a maximal isotropic subgroup of rank g of $A[2^{n+2}]$, $T'_1, \dots, T'_g$ generators of K', and we want to compute the isogeny $K = K'[2^n]$ with generators $T_i = 4T'_i$. As explained in the introduction, we will normalise once and for all the T'_i. The computations in Section 7 show that it is enough to completely normalise the points of 4-torsion in each K_i (up to a sign at the very last step when computing ϕ_n, but as always this sign does not matter because we only need the squares of these coordinates). So each T_i will give an equation $\lambda_i^{2^{n+2}} = C_i$, and we keep track of these normalisation factors at each step of our algorithm. Once again, from Section 7 we know that we will only need the values of the C_i and we never need to know the λ_i.

This global normalisation of K' is particularly useful when A is a product of g elliptic curves. Indeed, the normalisation procedure essentially boils down to a scalar multiplication (computed via a Montgomery ladder), and it is slightly faster to compute g such multiplications in dimension 1 than one in dimension g via the product theta structure. Furthermore, most cryptographic applications come from Kani's lemma, so that A is of the

form $E_1^{g/2} \times E_2^{g/2}$. So we really only need to normalise 4 points in dimension 1 (a basis of $E_i[2^{n+2}]$) rather than g^2, and then keep track of our normalisations across each copy of E_i.

11. Computing the isogenous theta null point

Let K be our kernel, assume that it is compatible with the theta null point on A, and that we have computed normalisation $\widetilde{P_i}$ of a basis $(P_1, \dots, P_g)$ of $A[4]$ (either from $K[8]$ or via a global normalisation). Let $B = A/K[2]$. We can use these normalised points to compute the correct choice of square roots for ${\theta'}^B_\chi(0)$. Let us first recall the formula from [DLRW23, § C.2] (which as already mentioned result from the duplication formula [Rob10, Théorème 4.4.3]), remembering that we need to conjugate them by H in our situation because here we consider the "dual" kernel on A.

In the original algorithm, we actually need $\widetilde{P_t}$ for any $t \in Z(\overline{2})$, where $P_t = \sum_{i=1}^g t_i P_i$. First use H to convert $\theta_i(\widetilde{P_t})$ to $\theta'_\chi(\widetilde{P_t})$, we then have:

$$ {\theta'}^B_{\chi_t} = \sum_\chi \theta'_\chi(P_t)^2 $$

where χ_t is the character dual to t.

Example 11.1. When $g = 1$, we have $\widetilde{T}_0 = (a, b)$ a lift of the theta null point $(a : b)$. We have ${\theta'}^B(0) = (A, B)$, with $A^2 = a^2 + b^2$, $B^2 = a^2 - b^2$ by Section 5. We have $T_1 = (1 : 0)$ (this is the only compatible point of 4-torsion above $(-a : b)$, the other one is $(0 : 1)$ and is not compatible as we will see shortly), and $\widetilde{T}_1 = (\lambda, 0)$ with $\lambda^4 = A^2B^2$. So $\theta'(\widetilde{T}_0) = (a + b, a - b)$, $\theta'(\widetilde{T}_1) = (\lambda, \lambda)$, and our formula above shows that ${\theta'}^B_0(0) = (a + b)^2 + (a - b)^2 = 2a^2 + 2b^2 = 2A^2$, and ${\theta'}^B_1(0) = \lambda^2 + \lambda^2 = 2\lambda^2 = 2AB$. The point $(2A^2 : 2AB) = (A : B)$, and we choose for affine lift on the dual theta null point of B the point $(1, B/A)$.

Given a point $P = (x : z)$, as explained in Section 5 its image by the isogeny in theta coordinates on B is given by $(x : z) \xrightarrow{S} (x^2 : z^2) \xrightarrow{H} (x' = x^2 + z^2 : z' = x^2 - z^2) \xrightarrow{C} (x'' = x'/A, z'' = z'/B) \xrightarrow{H} (x'' + z'', x'' - z'')$.

When working with projective coordinate, we only need the projective point $C = (1/A : 1/B)$. However when working with affine coordinates, since we want to send (a, b) to our choice of $(1, B/A)$, we need to take $C = (1/A^2, 1/AB)$. Let $(a_2 : b_2)$ be the theta null point on B.

We remark that T_1 is sent to $(-a_2 : b_2)$, the kernel of the next isogeny, while $(0 : 1)$ is sent to $(-b_2 : a_2)$, which is not the kernel of the next isogeny.

We now describe our optimisation. Let $i \in Z(\overline{2})$, and $\widetilde{T_i}$ be the corresponding normalised point of 4-torsion. Its image by our isogeny f has to be the normalised point of 2-torsion induced by i given in Section 3. Since this image is given by the operator $H \circ C \circ H \circ S$ with $C = 1/{\theta'}^B_i(0)$, this means that if we apply $C \circ H \circ S$ to $\widetilde{T_i}$, we obtain the point $({\theta'}^B_{\chi_i+\chi}(0))_\chi$, where χ_i is the character dual to χ. So $H \circ S(\widetilde{T_i}) = ({\theta'}^B_{\chi_i+\chi}(0){\theta'}^B_\chi(0))$.

In particular, applying this to all the T_i, we recover all two by two product $({\theta'}^B_\chi(0){\theta'}^B_{\chi'}(0))$, which gives an alternative way to recover the theta null point of B. But actually, it is enough to recover this theta null point by applying $H \circ S$ to only a basis $\widetilde{T}_1, \dots, \widetilde{T_g}$ along with the theta null point $\widetilde{T}_0$. Indeed, an explicit computation shows that we recover all ${\theta'}^B_\chi(0)/{\theta'}^B_0(0)$ for all characters χ of Hamming weight 1, then 2, and so on.

Example 11.2. When $g = 1$, we have $\widetilde{T_1} = (\lambda, 0)$ with $\lambda^4 = A^2B^2$. We apply $H \circ S$ to $\widetilde{T_0} = (a, b)$ to get (A^2, B^2), and to $\widetilde{T_1}$ to get $(\lambda^2, \lambda^2) = (AB, AB)$. From this we recover B/A, hence (a_2, b_2).

We remark that applying $H \circ S$ to $(0, \lambda)$ gives $(AB, -AB)$. In fact, for all sign choices of $(A : B)$, while $f(1 : 0) = (a_2 : -b_2)$, the kernel of the next isogeny, we have $f(0 : 1) = (b_2 : a_2)$.

The case $g = 1$ is particular in that we have some explicit points of 4-torsion in the theta model. So in that case, rather than looking at the preimage of our isogeny of the points of 2-torsion, we could look at the preimage of $(1 : 0)$. So let $T' = (r : s)$ a point of 8-torsion, this point fixes the (projective) theta null point of B. In particular, we should have $f(T') = (1 : 0)$. Doing the computation, we get $(r^2 + s^2 : r^2 - s^2) = (A : B)$. So in that case we can directly recover $(A : B)$ from T' in projective coordinates, without requiring any normalisation. The sign choice of $(A : B)$ induced from T' ensures that $f(T') = (1 : 0)$ becomes the compatible point of 4-torsion.

Example 11.3. When $g = 2$, let $(a_{00}, a_{01}, a_{10}, a_{11})$ be our theta null point on A, $(a'_{00}, a'_{01}, a'_{10}, a'_{11})$ our theta null point on B, and $(A_{00}, A_{01}, A_{10}, A_{11}) = H(a'_{00}, a'_{01}, a'_{10}, a'_{11})$ our dual theta null point on B. Recall that H is given by $H(x_{00}, x_{01}, x_{10}, x_{11}) = (x_{00} + x_{01} + x_{10} + x_{11}, x_{00} + x_{01} - x_{10} - x_{11}, x_{00} - x_{01} + x_{10} - x_{11}, x_{00} - x_{01} - x_{10} + x_{11})$.

Assume that we have normalised our points of 4-torsion $\widetilde{T_i}$. Recall that the isogeny is given by $f = H \circ C \circ H \circ S$ with $C = (1/A_i)$, and let $g = H \circ f = C \circ H \circ S$ the isogeny given in dual theta coordinates on B, and $h = H \circ S$ the isogeny given in twisted dual theta coordinates on B. We have $f(\widetilde{T_0}) = f(a_{00}, a_{01}, a_{10}, a_{11}) = (a'_{00}, a'_{01}, a'_{10}, a'_{11})$, so $g(\widetilde{T_0}) = (A_{00}, A_{01}, A_{10}, A_{11})$, and $h(\widetilde{T_0}) = (A^2_{00}, A^2_{01}, A^2_{10}, A^2_{11})$.

We know that $f(\widetilde{T_1}) = (a'_{00}, -a'_{01}, a'_{10}, -a'_{11})$, so $g(\widetilde{T_1}) = (A_{01}, A_{00}, A_{11}, A_{10})$, and $h(\widetilde{T_1}) = (A_{00}A_{01}, A_{00}A_{01}, A_{10}A_{11}, A_{10}A_{11})$.

We know that $f(\widetilde{T_2}) = (a'_{00}, a'_{01}, -a'_{10}, -a'_{11})$, so $g(\widetilde{T_2}) = (A_{10}, A_{11}, A_{00}, A_{01})$, and $h(\widetilde{T_2}) = (A_{00}A_{10}, A_{01}A_{11}, A_{00}A_{10}, A_{01}A_{11})$.

Finally, we know that $f(\widetilde{T_1 + T_2}) = (a'_{00}, -a'_{01}, -a'_{10}, a'_{11})$, so $g(\widetilde{T_1 + T_2}) = (A_{11}, A_{10}, A_{01}, A_{00})$, and $h(\widetilde{T_1 + T_2}) = (A_{00}A_{11}, A_{01}A_{10}, A_{01}A_{10}, A_{00}A_{11})$.

We see that the four points $\widetilde{T_0}, \widetilde{T_1}, \widetilde{T_2}, \widetilde{T_1 + T_2}$ allow to recover all 2 by 2 products A_iA_j. But the first three are already enough: dividing by A^2_{00}, we recover A_{01}/A_{00} from $\widetilde{T_1}$ and A_{10}/A_{00} from $\widetilde{T_2}$, which allows us to recover A_{11}/A_{00} from either of these two points.

Example 11.4. Assume that $g = 3$, and lets look at the image of the operator $h = H \circ S$, i.e, the isogeny f given in twisted dual theta coordinates on B.

We compute

$$h(\widetilde{T_0}) = (A^2_{000}, A^2_{001}, A^2_{010}, A^2_{011}, A^2_{100}, A^2_{101}, A^2_{110}, A^2_{111}),$$

$$h(\widetilde{T_1}) = (A_{000}A_{001}, A_{001}A_{000}, A_{010}A_{011}, A_{011}A_{010}, A_{100}A_{101}, A_{101}A_{100}, A_{110}A_{111}, A_{111}A_{110}),$$

$$h(\widetilde{T_2}) = (A_{000}A_{001}, A_{001}A_{011}, A_{010}A_{000}, A_{011}A_{001}, A_{100}A_{110}, A_{101}A_{111}, A_{110}A_{100}, A_{111}A_{101}),$$

$$h(\widetilde{T_3}) = (A_{000}A_{100}, A_{001}A_{100}, A_{010}A_{110}, A_{011}A_{111}, A_{100}A_{000}, A_{101}A_{001}, A_{110}A_{010}, A_{111}A_{011}).$$

Looking at the image of the $\widetilde{\sum \varepsilon_i T_i}$ we would also get all the 2 by 2 products A_iA_j, but these points are enough. We first recover A_{001}/A_{000}, A_{010}/A_{000}, A_{100}/A_{000}, then A_{011}/A_{000}, A_{101}/A_{000}, A_{110}/A_{000} and finally A_{111}/A_{000}.

12. The image of a point

We already saw how to compute the image of a point by the 2-isogeny f once we have the dual theta coordinates ${\theta'}^B_\chi(0)$ on B. Namely the formula is given by the operator $H \circ C \circ H \circ S$ where $C = 1/{\theta'}^B_\chi(0)$. This assume that these theta constants are non zero however.

In this section we explain how to deal with the annulation of some of these theta constants. This will typically be the case when the starting variety is a product of elliptic curves and the first isogeny a gluing isogeny.

Let $(\widetilde{T_1}, \dots, \widetilde{T_g})$ be our basis of normalised points in $K[4]$. Let P be a point on A, fix an arbitrary lift $\widetilde{P}$, and assume we have computed coherent lifts $\widetilde{P + T_i}$ relatively to $\widetilde{P}$ and the $\widetilde{T}_i$. Note that if $P \in K$ and we have already normalised all points in K, we can use these as normalisations.

The operator $h = H \circ S$ gives the image of P in terms of the twisted dual theta coordinates on B. In particular, if $\widetilde{Q} = f(\widetilde{P})$, we have $h(\widetilde{P}) = ({\theta'}^B_\chi(\widetilde{Q}){\theta'}^B_\chi(0))$, and for $i \in Z(\overline{2})$, $h(\widetilde{P + T_i}) = ({\theta'}^B_{\chi+\chi_i}(\widetilde{Q}){\theta'}^B_\chi(0))$. So we can use these points to recover all the coordinates of $h(\widetilde{P})$.

Example 12.1. When $g = 2$, and ${\theta'}^B_\chi(f(\widetilde{P})) = (x_{00}, x_{01}, x_{10}, x_{11})$, we compute $h(\widetilde{P}) = (A_{00}x_{00}, A_{01}x_{01}, A_{10}x_{10}, A_{11}x_{11}, h(\widetilde{P + T_1}) = (A_{00}x_{01}, A_{01}x_{00}, A_{10}x_{11}, A_{11}x_{10}, h(\widetilde{P + T_2}) = (A_{00}x_{10}, A_{01}x_{11}, A_{10}x_{00}, A_{11}x_{01}$, and $h(\widetilde{P + T_1 + T_2}) = (A_{00}x_{11}, A_{01}x_{10}, A_{10}x_{01}, A_{11}x_{00}$.

We see that even if one of the dual isogenous theta null point A_i is zero, knowing the (affine) theta coordinates of $P, P + T_1, P + T_2$ still allows to compute $h(P)$.

13. The full algorithm

Let us summarize the steps to compute a 2^n-isogeny with kernel K.

(1) Start with a theta null point of level 2 and A induced by some explicit symplectic basis $(e'_1, \dots, e'_g, f'_1, \dots, f'_g)$ of the 4-torsion. This can be done using Section 8 when A is a product of elliptic curves.
(2) Let $v'_1, \dots, v'_g$ be a basis of $K[4]$, and complete this basis into a symplectic basis $(u'_1, \dots, u'_g, v'_1, \dots, v'_g)$. Let M be the symplectic matrix $(e'_1, \dots, e'_g, f'_1, \dots, f'_g)$ to $(u'_1, \dots, u'_g, v'_1, \dots, v'_g)$. Apply the theta transformation formula induced by M to get the linear change of variable inducing new theta coordinates compatible with our kernel K.
(3) Let $T_1, \dots, T_g$ be a basis of K. For reasons explained in Section 9, it is convenient to assume that we are given T''_i an isotropic basis of $A[2^{n+2}]$ with $T_i = 4T''_i$. Using Section 7, normalise each T''_i to get an affine point $\widetilde{T''_i}$. If A is a product of elliptic curves, it will be easier and faster to normalise before the linear change of variable from the preceding step, because the normalisation for the product theta structure can be done in dimension 1.
(4) Compute $2^n\widetilde{T''_i}$ using Section 6, and use this normalised basis of $K[4]$ to compute the first isogeneous theta null point using Section 11.
(5) Compute the image of the $\widetilde{T''_i}$ using Section 12.
(6) Go back to step Item 4, with n decremented by 1.

We will also look at the variant where instead of normalising the points of 2^{n+2}-torsion T''_i at the beginning, we will only normalize points of 8-torsion at each step. In this variant we compute $U_i = 2^{n-1}T''_i$ to get g points of 8-torsion, which we normalise using Section 7. We then compute the affine 4-torsion point $2\widetilde{U_i}$ to recover the isogeneous theta null point

using Section 11. Then we compute the image of T_i'' using Section 12 as above, except that in this case we only need the projective image rather than the affine image since T_i'' is no longer normalised).

14. Complexity

Because of the dynamic nature of the algorithm optimising the number of isogeny images vs doubling, we need to plug parameters to compare algorithm. Still, we can do some naive complexity estimate to estimate the cost of the full isogeny computations with respect to the dimension.

14.1. **The old algorithm.** For the isogeny algorithm of [DLRW23], the naive ratio was given by $\kappa 2^g 2^g : 2^g$ points to track, each point using 2^g coordinates.

For the more refined estimation, we have the following complexities:

- Doubling a point still costs $2.2^g S + 2.2^g M = 4.2^g$ operation by points and computing the image of a point $2^g S + 2^g M = 2.2^g$ operations (without any inversions).
- Computing an isogenous theta null point costs $2^g(7.2^g - 2) - 2$ arithmetic operations (neglecting additions and soustractions). This involves computing the necessary inverse needed for the doublings and images of points.

The discrepancy with the more precise estimated ratio comes from the fact that computing the theta null points behave differently from computing the other $2^g - 1$ points.

2^n	$g = 1$	$g = 2$	$g = 4$	$g = 8$
2^{128}	8028	44328	850464	228774144
2^{216}	14476	80376	1546608	416370768
2^{250}	17060	94860	1826700	491877900
2^{305}	21350	118950	2292990	617612190
2^{372}	26576	148296	2861016	770779416
2^{486}	35904	200844	3879828	1045623348

g	Naive ratios	Estimated ratios
2	×4	×5.5
4	×64	×110
8	×16384	×29000

In these notes, we will try to minimise the number of inversions and divisions, since they are much more expensive than the other arithmetic operations (squares and multiplications). Also we will count one division as $1I + 1M$, so we will only track the number of inversions.

To normalize a point of 8-torsion T_i'', we compute $2T_i'', 3T_i''$. The computations of the theta coordinates of the T_i'' require some divisions: the duplication formula naturally give the $\theta_j(3T_i'')\theta_j(T_i'')$. But we don't actually need these divisions, $3T_i''$ is needed only for the normalisation constant, so we need just one of his coordinate. And we can compute this constant as $C = \theta_j(3T_i'')\theta_j(T_i'')/\theta_j(5T_i'')\theta_j(T_i'')$. Recall that $5T_i''$ is computed at $T_i'' + T_i$ where $T_i = 4T_i''$ is a point of 2-torsion (hence the translation is given by the explicit linear action of Section 3).

So the first doubling for $2T_i''$ (taking into account we are going to do a tripling) will cost $2^g S + 3.2^g M$ (using the fact that we precomputed the inverse of some theta constants). Then for computing one coordinate of $3T_i''$ we need $2^g S + 2^g M$. As an aside, this will compute

the square of the coordinates of the 4-torsion points $2T_i''$, which are needed to compute the isogenous theta null point.

We then compute C by the equation above, this costs $2M + 1I$. Using this constant to normalize our sum adds $1M$.

Since we normalize $2^g - 1$ points, the total cost to compute the isogenous theta null point is $(2^g - 1)(2.2^g S + 4.2^g M + 3M + 1I)$.

Note that here we don't take into account the cost of computing the squares of the theta null point, this was already done for the doubling computations. For the image of points we need to invert the (dual) coordinates of the isogenous theta null point, this costs $2^g I$.

We also need to compute $2^g S + 2^g I$ for the doubling operations on B, for computing the inverse of the square of the isogenous theta coordinates (it might seem that we would need $2^g I$ more to compute the inverse of the theta constants of B, but we already have the inverse of the dual coordinates, so we can compute the doubling in dual coordinates, and we just need the $1/\theta_i^A(0)^2$).

So the total cost to compute the theta null point and all inverse needed for images and doublings is $(2^g - 1)(2.2^g S + 4.2^g M + 3M + 1I) + 2^g S + 2.2^g I = 2^g(6.2^g + 1) - 4$ arithmetic operations, including $3.2^g - 1$ inversions.

With these new estimates, the above tables become:

2^n	$g = 1$	$g = 2$	$g = 4$	$g = 8$
2^{128}	8028	43560	823584	220483584
2^{216}	14476	79080	1501248	402380448
2^{250}	17060	93360	1774200	475685400
2^{305}	21350	117120	2228940	597857340
2^{372}	26576	146064	2782896	746684976
2^{486}	35904	197928	3777768	1014145128

g	Naive ratios	Estimated ratios
2	×4	×5.5
4	×64	×105
8	×16384	×28000

We will use these operations count to compare the new algorithm with the old one.

14.2. **The new algorithm.** To compute a 2^n-isogeny in dimension g, we need to keep track of the theta null points and of a basis T''_i. Then we only compute image of points, which cost roughly $2^g(S + M)$ by point, doublings, which cost roughly the same as 2 images, and the isogeneous theta null point, which cost roughly $1 + g$ images, g normalisations (which is roughly one doubling + one differential additino) along with some inverse to speed up the upcoming computations.

A rough estimate of the complexity to compute the isogenous theta constants is then around $\kappa(1 + g)2^g$, where κ will not depend too much on the dimension: we have $(1 + g)$ points to push and each point is represented by 2^g-coordinates. But for 2^n-isogenies we need to keep track of the basis, the optimal strategy uses a dynamic strategy due to [DJP14] optimising the number of doublings vs images according to their cost. This part of the algorithm can be estimated as $\kappa_2 g 2^g$: g points for the basis with 2^g coordinates each. In practice this part is dominating, it is roughly twice as expensive as the theta constant phase

(see below for more precise ratios). So we will use this to estimate our complexity ratio as our dimension increase.

A more refined estimate relies on using the optimal algorithm to choose between doubling points and pushing them by the isogeny.

14.3. **The new algorithm: normalising 8-torsion points at each steps.** Let us first begin with the case where we normalize at each step like the previous algorithm, rather than once at the beginning. This allows for a better comparison with the old algorithm, the difference being that we need to keep track of only g points (along with the theta null point), rather than $2^g - 1$.

(1) Doubling a point costs $2.2^gS + (2.2^g - 1)M = 4.2^g - 1$ operation by points and computing the image of a point $2^gS + (2^g - 1)M = 2.2^g - 1$ operations (without any inversion). The difference with the complexity of the old algorithm is that in this case we naturally compute the isogenous theta null point with the first coordinate normalised to 1.

(2) For each isogeny, we normalize points of 8 torsion (a basis of $K[8]$), we already saw above that this costs $2.2^gS + 4.2^gM + 1M + 1I = 6.2^g + 2$ by point (we gain $1M$ in our doubling because of the normalised theta constant).

We compute $1/A_0^2$ in $1I$, recover the $A_i/A_0, A_0/A_i$ for i of Hamming weight one in $1M + 1I$, and then the $A_i/A_0, A_0/A_i$ for the other i in $2M + 1I$ (write $A_i/A_0 = A_iA_j \times A_0/A_j \times 1/A_0^2$). Also each of these constants need to be normalised by the appropriate projective factor, this costs $(2^g-1)M$. The final cost is $1I+g(1M+1I) + (2^g - g - 1)(2M + 1I) + (2^g - 1)M = 4.2^g - g - 3$.

For our doubling on the isogenous variety, we need to precompute some constants, for a total of $2^gS + 2^gI$.

The total cost to compute the isogenous theta null point along with all the inverses needed for the doublings and images is thus of $g(6.2^g + 2) + 4.2^g - g - 3 + 2.2^g = 6(g + 1)2^g + g - 3$, including 2.2^g inversions.

We can plug these costs into the dynamic algorithm optimising the number of doublings vs images for a 2^n-isogeny. This gives us the following estimation of the number of arithmetic operations for computing 2^n-isogenies in different dimensions, we can also estimate the ratios and compare them with the naive expected ratios, and more importantly look at the efficiency gain compared to the previous algorithm.

2^n	$g = 1$	$g = 2$	$g = 4$	$g = 8$
2^{128}	7076	28032	224544	7099584
2^{216}	12704	50688	407976	12930264
2^{250}	14953	59776	481742	15277834
2^{305}	18663	74841	604437	19187709
2^{372}	23254	93340	754196	23951236
2^{486}	31275	126096	1022034	32500950

g	Naive ratios	Estimated ratios	Gain
1			×0.87
2	×4	×4	×0.64
4	×32	×32	×0.27
8	×1024	×1024	×0.032

If we look at the proportion of operations needed to compute the isogenous theta constants (along with all the constants needed for doubling and images), we see that depending on the isogeny size, this proportion is 35–40% for $g = 1$, 27–32% for $g = 2$, 22–27% for $g = 4$ and 20–25% for $g = 8$ (the longer the isogeny, the less the proportion).

We remark that once the chain of 2-isogenous theta null point is computed (along with the associated constants), we can compute the image of any point by our big 2^n-isogeny f or its dual in $n \times (2^g S + 2^g M)$ (we gain one M by image if our constants have been normalised to have one dual theta coordinate equal to 1).

14.4. **The new algorithm, normalizing points at the beginning.** Now we look at the complexity of the new algorithm, where we normalize points at the beginning and use affine images and doublings at every step. We treat here the case of a general kernel, already compatible with the theta null point.

(1) Normalising the point T_i'' costs a scalar multiplication to compute $(2^{n+1}-1)T_i'', 2^{n+1}T_i''$, and one more differential addition to compute $(2^{n+1}+1)T_i''$. The scalar multiplication costs 7.2^g arithmetic operations by bits (in the general case, it is slightly faster if we normalise one of the theta null coordinate to be 1). We need to compute the inverse of the theta coordinates of T_i'' first (for $2^g I$), and then we only need multiplications and squares afterwards. The extra differential addition costs 4.2^g arithmetic operations, but since this is only used to get the normalisation factor, as explained above, we actually only need $2^g S + 2^g M + 2M + 1I$.

Since we have g points to normalize, the normalisation phase costs $g \times ((n+1).7.2^g + 3.2^g + 3)$, including $g(2^g + 1)$ inversions.

(2) Keeping track of the normalisation factor. Recall that we have an equation $\lambda_1^{2^{n+2}} = C$ for our each of our normalisation factor. To compute the points of 4-torsion from the T_i'', we need to adjust our points by the normalisation factor $\lambda_1' = \lambda_1^{2^{2n}}$. For computing the theta null point, we only need ${\lambda_1'}^2 = \lambda_1^{2^{2n+1}} = C^{2^{n-1}}$.

When we compute the image of T_i'', the new normalisation factor is $\lambda_2 = \lambda_1^2$. We have $\lambda_2^{2^{n+1}} = \lambda_1^{2^{n+2}} = C$. so for the second isogeny, the normalisation factor on the points of 4-torsion is then $C^{2^{n-2}}$ and so on, until at the last step we use C. In total we need $n-1$ squares to compute the actual constants which will give us our normalisation factors for each of our isogenies. Since we have g normalisation factors, this adds $g(n-1)S = g(n-1)$ arithmetic operations.

(3) Doubling a point costs $4.2^g - 1$ operation by points (we are in a situation where one of the theta constant is normalised to 1).

(4) Computing the image of a point costs $2.2^g - 1$ operations.

(5) Computing an isogenous theta null point require computing the squares of the domain theta null point and the g points of 4-torsion forming a basis of $K[4]$; this costs $(1+g)2^g$ squares.

We need to compute the A_i/A_0 (for the isogenous theta null point), $1/A_0A_i$ (for the isogeny images, unlike the previous algorithm where we used A_0/A_i here we need to use the affine isogeny formula, so correct by the renormalisation we used for the theta null point), and the A_0/A_i (for doubling in the dual theta coordinates).

We can compute them as follow: first compute $1/A_0^2$, then compute $A_i/A_0 = A_iA_j/A_jA_0$, $A_0/A_i = 1/(A_i/A_0)$, $1/A_0A_i = A_0/A_i \times 1/A_0^2$ for a total cost of $1I + 2M$ by coefficient. We need to add $1M$ to take into account the normalisation factor. Thus computing the isogenous theta null points costs $1I + (2^g - 1)(1I + 3M) = 4.2^g - 3$ operations.

For doubling (in dual theta coordinates) on the isogenous abelian variety, we need the coefficients A_0^2/a_i^2 (because we need to do affine doublings and we renormalised our theta null point), this costs $2^g I + 2^g M$.

The grand total to compute the theta null points and all inverse needed for images and doublings is $(1+g)2^g + 4.2^g - 3 + 2.2^g = 2^g(g+7) - 3$, including including $2.2^g I$.

In summary, the amortised cost for computing an isogenous theta null point (taking into account the normalisation at the beginning) is of $g.7.2^g + 2^g(g+7) - 3 = 2^g(7g+g+7) - 3$ arithmetic operations, including $2.2^g I$.

The amortised cost for the normalisation is roughly a doubling and differential addition for each of the g points in our basis. So this is about the same cost as we obtain by normalising the points of 8-torsion anew for each isogeny, except that in the latter case we don't need a full differential addition, only a partial one, and we can compute doublings and images projectively since we don't need to keep track of our normalisation factors because we recompute them at each step.

Thats why, normalising at each steps gives better complexity. But note that for cryptographic applications where $A = E_1^g \times E_2^g$, we could just normalize a basis of $E_i[N]$, and then switch to affine differential additions when computing $\mathrm{Ker}\, F$ to keep points normalised. This allows to normalize only 4 points in dimension 1, instead of $2g$ points in dimension $2g$ (which amount roughly to normalizing $4g^2$ points in dimension 1). This gains a factor roughly 4 when $g = 2$, i.e., $\dim A = 4$ (roughly because once the points are normalised, to compute the kernel of F we need to use affine differential additions rather than projective ones, this will cost $1M$ more). So we expect that for the cryptographic setting of 2^n-isogenies in dimension 4, the method of normalising points globally will be more effective, because we will be able to do the normalisation in dimension 1.

The estimated number of operations is summarised in the following table, note that here these operations count do not assume that the initial variety is a product, so we compute the normalisation in dimension g.

2^n	$g=1$	$g=2$	$g=4$	$g=8$
2^{128}	7866	30676	243624	7677136
2^{216}	14022	55092	439728	13890792
2^{250}	16475	64860	518390	16386330
2^{305}	20515	81025	649005	20535565
2^{372}	26576	146064	2782896	746684976
2^{486}	35904	197928	3777768	1014145128

g	Naive ratios	Estimated ratios	Gain
1			×0.95
2	×3	×4	×0.7
4	×20	×32	×0.29
8	×576	×1000	×0.034

15. 2^n-ISOGENIES IN DIMENSION 1

15.1. **2-isogenies in the theta model.** The above algorithm is generic and work in any dimension; the resulting number of operations in dimension 1 simply amount to plugging $g = 1$ in the formula.

However in practice computing 2^n-isogenies in the theta model in dimension 1 is faster than the generic algorithm, because we can dispense with point normalisation in dimension one. (Update: see Section 17 for the same tricks in higher dimension.)

The algorithm is thus as follow: let $0_E = (a : b)$ be the theta null point, given by a theta structure such that our small kernel is generated by $T = (-a : b)$.

We assume that we have a point of 8-torsion T'' above T, if $n \geq 3$ we want $T'' \in K$, and if $n = 2$ we want $2T''$ in K, this ensure that if our isogenous theta null point on E_2 is $(a_2 : b_2)$, our next kernel will be generated by the point $(-a_2 : b_2)$.

The formula are sufficiently simple that we will also keep track of the additions/soustractions.

Let $T'' = (r : s)$, we have by Example 11.2 $(A : B) = (r^2 + s^2 : r^2 - s^2)$, and $(a_2 : b_2) = (A + B : A - B) = (r^2 : s^2)$. This requires $2S + 2a$.

The image of a point $P = (x : z)$ is given by $(x : z) \mapsto (x^2 : z^2) \mapsto (X = x^2 + z^2 : Z = x^2 - z^2) \mapsto (BX : AZ) \mapsto (x' = BX + AZ : z' = BX - AZ)$, and is computed in $2S + 2M + 4a$.

Doubling on E_2 cost two images, the first one for the dual isogeny $\tilde{f}$ using the coefficients $(a : b)$ instead of $(A : B)$ in the formula above, and the second using f to go back to E_2, for a total cost of $4S + 4M + 8a$.

If we have many doublings and images to compute, it might be worth to compute $(1, B/A)$. This can be done with one division, that is $1I + 1M$. We then gain $1M$ for images and doubling. At this point we might as well compute also $(1, b/a)$. We can compute both $1/a, 1/A$ in $1I + 3M$, so compute $(1, b/a), (1, B/A)$ in $1I + 5M$. We then gain $2M$ for images and doubling. In summary, adding one inversion, computing the normalised theta null points and associated constant costs $1I + 5M + 2S + 2a$ (instead of $2S + 2a$), and the computing an image costs $2S + 1M + 4a$ and a doubling $4S + 2M + 8a$ (instead of $2S + 2M + 4a$ and $4S + 4M + 8a$ respectively).

We obtain the following costs in dimension 1 (which give roughly a twenty percent speedup compared to the generic algorithm, not counting the fact that here the arithmetic operations are without any inversion).

2^n	$g = 1$
2^{128}	5468
2^{216}	10156
2^{250}	12060
2^{305}	15250
2^{372}	19136
2^{486}	26184

If we don't have an available point of 8-torsion T'', we simply compute $A^2 = a^2 + b^2, B^2 = a^2 - b^2$ and take an arbitrary square root of B^2/A^2. What we can do also, without requiring a square root, is to compute the codomain in the Montgomery model, see the next section.

15.2. **Theta versus Montgomery.** To summarize, the complexities for computing isogenies in the theta model are as follows:

(1) $2S + 2a$ for the codomain
(2) $2S + 2M + 4a$ for an image
(3) $4S + 4M + 8a$ for doubling

The input is the theta null point $(a : b)$, which implicitly contains the 2-torsion point $(-a : b)$ used for our kernel; and the images computations needs (some constants computed during) the codomain. We refer to [Rob23b] for similar formulas on twisted theta models.

In the Montgomery model, the costs are, using [CLN16; CH17; Ren18]:

(1) $2S + 1a$ for the codomain
(2) $4M + 4a$ for an image (using a precomputation of $2a$)
(3) $2S + 4M + 4a$ for doubling

Here the input is a two torsion point (different from $(0 : 1)$) giving the kernel (and implicitly the curve); the image computation does not needs the codomain.

In [RS24] (see the notes [Rob23b] for more on the arithmetic of Kummer lines), we explain how to combine the best of both worlds. Provided we have a point of 4-torsion T above our kernel $\langle T_1 \rangle$, we can:

(1) Compute a representation of the codomain in $2S$. The representation is given by the 2-torsion point $f(T) = T_2$, which is the kernel of the next isogeny.
    If we need to compute doublings on the codomain, we need to add a $2S + 2a$ precomputation to compute $(\mathcal{A} + 2 : 4)$, and if we need to compute images we need to add a $2a$ precomputation (which is already done if we did the previous $2S + 2a$ precomputation needed for doublings).
(2) Compute "images" in $2M + 2S + 4a$.
(3) Compute "doublings" in $4M + 2S + 4a$.

The words "images" and "doublings" are in quotes because if we consider that we are on a twisted theta models the "doublings" we compute are actually $2P + T_1$, while if we consider that we are in the Montgomery model it is the images that are actually given by $f(P) + T_2$. The images need some of the constants computed for the codomain.

As an aside, we can also explain how to compute the isogeny from a theta model to a Montgomery model if we do not have access to a 8-torsion point. From the theta null point $(a : b)$ of E_1, we can compute $(a^2 : b^2)$ the theta null point of E_2 in the $\theta^2_{E_1} = \theta' tw'_{E_2}$ model, and the isogeny map is $(x : z) \mapsto (x^2 : z^2)$. Translating by $T_2 = (1 : 0)$ we obtain the coordinates on the Montgomery model of E_2, with $\mathcal{A}_2 = -\alpha_2 - 1/\alpha_2$, $\alpha_2 = b^2/a^2$.

We conclude this with a discussion on 4-isogenies. On the Montgomery model, a 4-isogeny can be computed in [CH17]:

(1) $4S + 5a$ for the codomain
(2) $6M + 2S + 6a$ for images.

Using these formula, it is faster to split a 2^n-isogeny in dimension 1 into blocks of 4-isogenies rather than blocks of 2-isogenies.

We leave as an open question the task of generalising these efficient 4-isogenies formula to the theta model in dimension 1 (or even better in higher dimension).

16. 2^n-isogenies in dimension 2

16.1. **Isogeny formula.** The estimation above are very rough because we count an inversion as much as a square or a multiplication. In this section we detail the detail the case of $g = 2$, and we try to use Montgomery's trick as much as possible to reduce the number of inversions needed by isogeny. Recall that this trick replace m parallel inversions by $1I + 3(m-1)M$.

For $g = 2$ we normalize our basis (P_1, P_2) of 8-torsion of $K[8]$ by one doubling, which cost (by point) $2.2^g S + (2.2^g - 1)M = 8S + 7M$, and then a partial differential addition which costs $2^g S + 2^g M + 2M + 1I = 4S + 6M + 1I$. Since we have two points, we can replace $2I$ by $1I + 3M$. The total cost is then $1I + 2.(8S + 7M + 4S + 6M) + 3M = 1I + 24S + 29M$.

These operations already give us the squares of the coordinates of the points of 4-torsion $2P_1, 2P_2$, and if we add the squares of the theta constants, we obtain by Example 11.3 (A^2, B^2, C^2, D^2), (AB, AB, CD, CD), (AC, BD, AC, BD) (up to the projective factors computed above) via $2^g S = 4S$. We need to add $3M$ to take into account the correcting factors for the coefficients AB, AC, BD, so the total cost is $4S + 3M$. We can exploit the fact that we work with projective coordinates to dispense with the $1I$ in the computation of the normalisation factor. If we don't compute this inversion, what we obtain are the points $(A^2, B^2, C^2, D^2, \kappa AB, \kappa CD, \kappa AC, \kappa BD)$ where κ is the element we did not inverse (which is the product of all elements we needed to inverse in parallel and which is computed as part of Montgomery's trick). So via $4M$, we can recover the projective vector $(A^2 : B^2 : C^2, D^2 : AB : AC : BD)$, we actually won't need C^2 so we just need $3M$. The final cost for this vector, in order to gain our $1I$, is $4S + 6M$.

We want to compute the isogeneous dual theta null point $(1 : B/A : C/A : D/A)$, and also for the image of the points the constants $(1 : A/B : A/C, A/D)$. We compute $1/A^2, 1/AB, 1/AC, 1/BD$. We recover $B/A = AB * 1/A^2, A/B = 1/AB * A^2, C/A = AC * 1/A^2, A/C = 1/AC * A^2, D/A = 1/BD * B/A * D^2, A/D = 1/BD * A/B * B^2$ in $4I + 8M = 1I + 17M$. In fact, using the same trick as above, we can entirely dispense with the inversion. If we do we obtain the coordinates $\kappa' B/A, \kappa' A/B, \kappa' C/A, \kappa' A/C, \kappa'^2 D/A \kappa'^2 A/D$ where κ' is the product of all coordinates we inverted. To recover the projective vectors $(A : B : C : D)$ and $(1/A : 1/B : 1/C : 1/D)$ we thus need to compute κ'^2 and do $4M$. This adds $1S + 4M$, for a cost of $1S + 21M$.

For the doubling on the isogenous abelian variety, we need to inverse 4 coordinates (the 4 squares of the theta constants needed are already taken into account above), for $4I$, i.e., $1I + 9M$. In this case, the inversion is not needed, since the projective factor will be the same.

However, while this is ok for projective doubling, for the affine doubling and differential addition we need when computing $3.T_i''$ for the normalisation, we will be off by some projective factors. Namely, since I compute the vector $(1/A : 1/B : 1/C : 1/D)$ up to some factor κ_1, and the vector $(1/a^2 : 1/b^2 : 1/c^2 : 1/d^2)$ up to some projective factor κ_2, the first doubling is off by a factor $\kappa_1\kappa_2$, and then the differential addition is off by a factor $(\kappa_1\kappa_2)^3$. This constant is computed via $2M + 1S$, and we need to use it for our 2 normalisation which adds $2M$.

The final cost is $(24S + 29M) + (4S + 6M) + (1S + 21M) + (9M) + (1S + 4M) = 30S + 69M \leq 99M$. In this case, the (dual) theta null point (A, B, C, D) is not normalised to have $A = 1$, so the image of a point then costs $4S + 4M$, and doubling costs $8S + 8M$.

If we have many doublings and images to compute, it might be interesting to add back $1I$ to normalise our coefficient A to be $A = 1$, and while we are at it $a = 1$. The image of a point then costs $4S + 3M$, and (projective) doubling costs $8S + 6M$.

We obtain the following number of arithmetic operations for our isogenies, without any inversion. We see that replacing all inversions by multiplication roughly augment the arithmetic count by twenty percent compared to the previous table, which is mainly due to the fact that our images are 15% slower ($4S + 4M$ vs $4S + 3M$) and our doublings 5% slower ($8S + 8M$ vs $8S + 7M$); the remaining cost being due to the fact that the isogenous theta constant and the associated constants needed for images and doublings take more arithmetic operations when we remove all inversions. The proportion of operations related to computing the isogenous theta null points (and associated constants for doubling and images) compared to doublings and images is between 32–37%.

2^n	$g = 2$
2^{128}	33520
2^{216}	60280
2^{250}	70990
2^{305}	88755
2^{372}	110396
2^{486}	148962

Remark 16.1. In our estimation of roughly $100M$ to compute the theta null point, half of it ($24S + 29M$) is spent normalizing our two points of 8-torsion (P_1, P_2). The normalisation computes (affinely) $2P_i, 3P_i$. At the next isogeny step, say we have for 8-torsion points on B the points (P'_1, P'_2). Then we have $2P'_i = f(P_i)$ projectively. So we can replace a doubling by an image (which is twice as fast, and in fact when computing $3P_i$ we essentially compute $f(P_i)$ along the way), and we just need one affine coordinate of $2P'_i$ to correct $f(P_i)$ to obtain the correct affine lift of $2P'_i$.

In other words, we can reuse part of the work of normalising our 8-torsion points on A to speed up normalising our 8-torsion points on B.

16.2. **Splitting isogenies.** In the contest of cryptography, the last 2-isogeny will be a splitting $A \to E_1 \times E_2$. During the isogeny computation, we will not in general obtain a product theta structure on $E_1 \times E_2$. In [DLRW23, Appendix C.1] we explain how, if we have enough information, we can precompute (by working in dimension 1) the linear change of variable giving a product theta structure.

But in dimension 2 it is easy to obtain it directly. First we know that we are on a product when one of the 10 even level $(2,2)$-theta constant is zero, and we know that we have a product theta structure where the zero theta constant is $\theta[11;11]$.

We might as well take a random linear change of variable induced by a symplectic action until we are on this case. A more deterministic algorithm (using Appendix B to stay in level 2) is as follows:

(1) The square of the level $(2,2)$ theta functions can be computed from the level 2 theta function via (this is a special case of the duplication formula) $U^2_{\chi,i} = \sum_t \chi(t)\theta_t\theta_{i+t}$.
    Suppose that we have a theta null point $(a : b : c : d)$ on a product. Let (χ, i) be the coordinate of the even theta constant which is zero.
(2) if $\chi = i = (00)$, act by $(a : b : c : d) \mapsto (a : ib : c : id)$, the new zero level $(2,2)$-theta function is given by (χ', i') with $\chi' \neq 0, i' = 0$.
(3) if $i = 0$ but $\chi \neq 0$ uses the action by H, this permutes χ and i.
(4) We can now assume $i \neq 0$. Take any invertible matrix A such that $A(11) = i$. Acting by $\theta_i \mapsto \theta_{Ai}$ we get that the new zero theta function is (χ', i') with $\chi' = \chi o A$, and $i' = (11)$.
(5) We can now assume $i = (11)$. Act by $\theta_i \mapsto (-1)^{(1-\chi)(i)}\theta_i$. The new zero theta function is given by (chi', i')=(1 1, 1 1) and we have won.
(6) If we have a point $(x : y : z : t)$ on the product theta structure, the theta coordinates on E_1, E_2 are given by $(x : z), (x : y)$.

16.3. **Gluing isogenies.** When we start with a product of two elliptic curves $E_1 \times E_2$ and a product theta structure $(a : b : c : d)$, then since the dual theta coordinates on the isogeneous surface $(A : B : C : D)$ can also be interpreted as the level $(2,2)$-theta coordinates on

the original surface given by $U_{\chi,0}(0)$, they are not zero (because the one which is zero corresponds to $\chi = i = (11)$).

However, if we take an isogeny which is not a diagonal isogeny, we will do a linear change of variable as explained in Sections 8 and 9 and one of the A, B, C, D will become zero.

For the arithmetic on $E_1 \times E_2$ this is not a problem (we would compute the arithmetic in dimension 1 before taking the product theta structure and doing the linear change of variable anyway), but this is a problem for the images of a point. The solution is given in Section 12: to compute $f(P)$, we need one of $P + T_1$, $P + T_2$ for T_1, T_2 the 2-points of 4-torsion compatible with our isogeny.

We note that if we only have P and T_1, there are four choices for $P + T_1$ on the Kummer. This corresponds to the fact that the map $E_1 \times E_2 \to E_1/\pm 1 \times E_2/\pm 1$ has degree 4, and given a point $[P] \in E_1/\pm 1 \times E_2/\pm 1$ we have 4 possibilities for P on $E_1 \times E_2$, which induces 4 possibilities on the codomain B, which induces 2 possibilities on $B/\pm 1$. So $[P]$ has two possible images on $B/\pm 1$, and we need a point of 4 torsion to fix one.

If our big kernel K is generated by P_1, P_2 of 2^n-torsion, we can take $T_i = 2^{n-2}P_i$. To compute our isogenies we need $f(P_1), f(P_2)$. But $P_i + T_i = (1 + 2^{n-2})P_i$ which can be computed via a scalar multiplication.

More concretely write $P_i = (R_i, S_i)$, then the four possibilities for $P_i + T_i$ can be written as $(1 \pm 2^{n-2}R_i, 1 \pm 2^{n-2}S_i)$. If we make some choice of sign for P_1 (say $(+,+)$) it is important to make the same for P_2 (say $(+,+)$ or $(-,-)$ but not $(+,-)$ or $(-,+)$) for our images of P_1, P_2 to be compatible. (The four choices for $P_1 + T_1$ corresponds to replacing P_1 by $-P_1$ or f by $-f$ in Kani's lemma. It might seem that we would need to fix $T_1 + T_2$ in order to fix the sign of P_1 relatively to P_2, but this is already done, at least implicitly, in our linear change of variable from our product theta structure: for this theta structure the basis of 4-torsion is of the form $(U_1, 0), (0, U_2), (V_1, 0), (0, V_2)$ which are points that only admits 2 preimages on $E_1 \times E_2$).

16.4. **Annulation of the theta null points.** Analytically, if A corresponds to the period matrix Ω, we have $(a, b, c, d) = \theta[0, i/2](0, \Omega/2)$ and $(A, B, C, D) = \theta[i/2, 0](0, \Omega)$.

The 10 level four even theta constants are $\theta[i/2, j/2](0, \Omega)$ are non zero, except when A is a product where exactly one of them is zero. And if A has a product theta structure, the zero even theta constant is $\theta[1/21/2; 1/21/2](0, \Omega)$.

From this we deduce that:

- (A, B, C, D) are non zero, except if A is a product with a non product theta structure.
- (a, b, c, d) are non zero, except if the isogenous abelian variety A/K_2 corresponding to $\Omega/2$ is a product with a non product theta structure.

So unless we encounter a product along our path (very unlikely), the only annulation we will see is at the first and last isogeny.

16.5. **Further optimisations in dimension 2.** Due to the ongoing work on implementing the formula in dimension 2, it is now easier to find new optimisation possibilities.

The image of a point is pretty fast, so the remaining bottleneck is to try to compute the theta constants as fast as possible.

There are two optimisations: first the normalisation procedure, a lot of the computations can be shared. Secondly, as remarked by Pierrick Dartois, the points of 4-torsion we deal with have 2 zero coordinates, so this simplify the computations.

First let's explain look at the points of 4-torsion: we have $T' + T = T'$ in the Kummer, where $T = 2T' \in K_2$. Since T acts by sign, this equation gives that half of the coordinates

are zero (we can even know which ones should be zero since we require the compatibility with the theta structure).

Secondly, let's look at the normalisation procedure. We have T'' a point of 8-torsion, and we compute $3T''$ to compute the correct projective factor λ. Note that we only need to apply this factor to $\theta'(f(T'))$.

Now from Section 5, our first (affine) doubling $T' = 2T''$ can be written as $\theta_i(T')\theta_i(0) = H(\theta'_i(T'')^2)$. As explained in Section 16.1, a doubling is $8M + 8S$ (if the appropriate inverses have been computed), but since we have two zero coordiantes the cost reduces to $6M + 8S$.

The main gain we can have is for the differential addition $3T'' = T'' + T'$, remember that $\theta_i(3T'') = \theta_i(5T'') = \theta_i(T'' + T)$ hence is equal to $\theta_i(T'')$ up to an explicit sign.

In particular, by the duplication formula, we have $\theta'_i(f(T''))\theta'_i(f(T')) = H(\theta_i(3T'')\theta_i(T''))$. Now we have already mostly computed $\theta'_i(f(T''))$ and $\theta_i(T'')^2$ during the doubling. To compute $\theta'_i(f(T'))$ usually requires $4S + 4M$, but the multiplication by the required constants can already be done during the doubling of T'', and the $4S$ is a $2S$ because two coordinates are zero. However this changes the ordering of operations for the doubling, which now costs $10M + 4S$ rather than $6M + 8S$.

So $\theta'_0(f(T'))$ requires (essentially) $2S$, and $\theta'_0(f(T''))\theta'_0(f(T'))$ adds $1M$. The correcting factor is then one division D, which we use to multiply two coordinates (because T' give half of the coordinates we are interested in), for a cost of $2M$. And in fact for the second generator, we just need one coefficient of $f(T')$ so we just add $1M$ for the correction.

In total, we have spent $(10M + 4S) + (2S + 1M) + 1D + 2M = 13M + 6S + 1D$ to get the correct affine value of $f(T')$. Doing this twice (once for each projective factor), we get (AB, CD, AC) with a cost of $25M + 12S + 2D$. Since we already know the value of (A^2, B^2, C^2, D^2) (since they were used for doubling; if we count them as precomputed then we need to add the computation of $(A_2^2 : B_2^2 : C_2^2 : D_2^2)$ as required precomputations for our theta null point), we recover as in Section 16.1 the values $(A : B : C : D)$, and then $(a_2 : b_2 : c_2 : d_2)$.

By contrast, the method outlined in Section 16.1 was costing $32M + 24S + 2I$ for the same result. We gain about 19 arithmetic operations.

We also need for the images $(1/A : 1/B : 1/C : 1/D)$ (which can be done in $4M$ because we already have $(1/A^2 : 1/B^2 : 1/C^2 : 1/D^2)$ and for doublings on the isogenous curve $(a_2^2 : b_2^2 : c_2^2 : d_2^2)$ to compute $(A_2^2 : B_2^2 : C_2^2 : D_2^2)$, $(1/A_2^2 : 1/B_2^2 : 1/C_2^2 : 1/D_2^2)$ and $(1/a_2 : 1/b_2 : 1/c_2 : 1/d_2)$. This requires $4S + 4M + 8I$.

A trick is to instead do doubling in θ' coordinates; for that we need $(1/A : 1/B : 1/C : 1/D)$ which we already have for images, and $(1/a^2 : 1/b^2 : 1/c^2 : 1/d^2)$. So from this point of view, anticipating the next isogeny, we need $(a_2^2 : b_2^2 : c_2^2 : d_2^2)$ and their inverse, so this does not change much the number of operations: $4S + 8I$, so we save $4M$.

For this part, we refer to Section 16.1; we can apply the same various M/I tradeoffs to get rid of all inversions at the cost of more multiplication. Remark 16.1 also apply, at the next step we could compute an isogeny image rather to speed up the doubling procedure.

16.6. **What if we don't have 8-torsion points?** If we only have points of 4-torsion T'_1, T'_2 above our kernel K_2, applying $h := H \circ S$ to them gives $(AB : AB : CD : CD)$, $(AC : AC : BD : BD)$. We also have $(A^2 : B^2 : C^2 : D^2)$ from the theta null point.

We cannot recover the theta null point $(A : B : C : D)$ directly because we are in projective coordinates. We can normalize T'_i via the equation $2.T'_i = T_i$, this determines the normalisation factor λ_i up to an equation $\lambda_i^4 = C_i$, hence this we square the coordinates, $h(T'_i)$ up to a sign. Hence we have 2 signs; and by the same method in dimension g we would have g signs, which are all valid by Appendix B.2. In particular, in dimension 2 we need

two square roots to compute the codomain theta null point when we only have points of 4-torsion and not of 8-torsion. In fact, we can rewrite the normalisation process as follow: let $h(T_1') = (\lambda_1 x, \lambda_1 x, \lambda_1 y, \lambda_1 y)$ for some unknown projective factor λ_1. Fix a choice of (A^2, B^2, C^2, D^2). Then for the correct choice of λ_1, we should have $\lambda_1 x = AB, \lambda_1 y = CD$, and we have an equation $\lambda_1^2 x^2 = A^2 B^2$, which gives λ_1 from a square root computation. The same method works for λ_2. If $(A : B : C : D)$ is one of the computed isogeneous theta null point, the other choice of signs give $(A : -B : C : -D)$, $(A : B : -C : -D)$, $(A : -B : -C : D)$.

If we don't even have the compatible points of 4-torsion above the kernel, we can write down equations which determines T_1', T_2', or even just $h(T_1'), h(T_2')$ which is what we really need for the codomain. Since T_1' is a compatible point of 4-torsion, and $T_1' + T_1 = T_1'$, we have $T_1' = (x : 0 : z : 0)$. From $h(T_1') = (AB : AB : CD : CD)$ we obtain a degree 2 homogeneous equation in x^2, z^2 (say $h(T_1') = (x_1 : x_1 : z_1 : z_1)$ with x_1, z_1 homogeneous of degree 2 in x, z, then $C^2D^2x_1^2 - A^2B^2z_1^2$), so we have two solutions for $h(T_1')$ (which is linear in x^2, z^2). Now $T_2' = (u : v : 0 : 0)$, and write $h(T_2') = (x_2 : z_2 : x_2 : z_2)$. We have $1/A^2 x_1 x_2 - 1/D^2 z_1 z_2$. This equation determines the projective point $h(T_2')$ uniquely from x^2, z^2. To our choice of signs above, this adds the possibility $(A : B : C : -D)$.

17. Even better formula: getting rid of the normalisation process

We have seen that in dimension 1 (Example 11.2) we don't need to normalize points of 8 and 4-torsion. So why do we need to normalize points in higher dimension? The answer is that we actually don't need this, which leads to both faster formula and much easier implementations.

The basic idea is as follow. Let's work in dimension $g = 2$ for simplicity. Let P_1, P_2 be a basis of $K[8]$, then we know that applying $S \to H \to C$ gives the images $f(P_1), f(P_2)$ in θ' coordinates.

Our kernels are set up so that $K[2] = K_2$, in particular $4P_1, 4P_2$ acts by sign. So in θ' coordinates, $2f(P_i)$ acts by permutation. $f(P_1), f(P_2)$ are points of 4-torsions, and since we are on the Kummer, we have $f(P_i) + 2f(P_i) = f(P_i)$.

Recall that $\theta'(f(P)) = g(P) := C \circ H \circ S$. This means that $g(P_1) = (x_1 : x_1 : z_1 : z_1)$ and $g(P_2) = (x_2 : z_2 : x_2 : z_2)$. Going one step back in the isogeny image formula, it means that if we apply $h := S \to H$ to P_1, P_2 (remember that $C = (1/A : 1/B : 1/C : 1/D)$ is unknown for now), we have $h(P_1) = (Ax_1 : Bx_1 : Cz_1 : Dz_1)$ and $h(P_2) = (Ax_2 : Bz_2 : Cx_2 : Dz_2)$ where x_1, z_1, x_2, z_2 are unknown projective factors.

But from this we can recover $B/A, C/A$ and $D/A = D/C \times C/A$ in only $2\times 4S + 1M + 3D$. Actually, for isogeny images we need $(1/A : 1/B : 1/C : 1/D)$, and we can compute $(1, A/B, A/C, A/D)$ in $2 \times 4S + 1M + 3D$. The nice thing is that a constant is normalised to 1, so images only cost $4S + 3M$. Then doublings could be implemented as composing $\hat{f}$ with f. For that we need $(1/a : 1/b : 1/c : 1/d)$, or better $(1, a/b, a/c, a/d)$; each doubling would cost $8S + 6M$.

As for doubling precomputations, for the next isogeny we would need to compute $(1/a_2 : 1/b_2 : 1/c_2 : 1/d_2)$ (or better $(1, a_2/b_2, a_2/c_2, a_2/d_2)$) from $(1, A/B, A/C, A/D)$. This can clearly be done through $3I + 3D$, but maybe there are some optimisations to be gained.

In higher dimension, the same strategy as in Section 11 holds. Let's work out the case $g = 3$, we have for P_1, P_2, P_3 a basis of $K[8]$, $g(P_1) = (x_1A_{000}, x_1A_{001}, y_1A_{010}, y_1A_{011}, z_1A_{100}, z_1A_{101}, t_1A_{110}, t_1A_{111})$, $g(P_2) = (x_2A_{000}, y_2A_{001}, x_2A_{010}, y_2A_{011}, z_2A_{100}, t_2A_{101}, z_2A_{110}, t_2A_{111})$, $g(P_3) = (x_3A_{000}, y_3A_{001}, z_3A_{010}, t_3A_{011}, x_3A_{100}, y_3A_{101}, z_3A_{110}, t_3A_{111})$.

So from the $g(P_i)$, which we can compute, we can recover the quotients $A_{001}/A_{000}, A_{010}/A_{000}, A_{100}/A_{000}$, and then iteratively $A_{011}/A_{000} = A_{011}/A_{010} \times A_{010}/A_{000}$, $A_{110}/A_{000} = A_{110}/A_{100} \times A_{100}/A_{000}$, $A_{111}/A_{000} = A_{111}/A_{110} \times A_{110}/A_{000}$.

In dimension g, computing the $g(P_i)$ costs $g \times 2^g S$, and then reconstituting the A_i/A_0 for isogeny images costs at most $2^g \times (1M + 1D)$, to which we need to add $2.2^g I$ for the arithmetic precomputations. So the total cost for the codomain, including the arithmetic precomputation, is $2^g(g + 4)$ arithmetic operations.

It is time for our table counting the number of arithmetic operations: we count each image as costing $2^g S + (2^g - 1)M = 2.2^g - 1$, and each doubling $2.2^g S + 2.(2^g - 1)M = 4.2^g - 2$. We have seen that the codomain and arithmetic precomputation costs $2^g(g + 4)$.

2^n	$g = 1$	$g = 2$	$g = 4$	$g = 8$
2^{128}	5189	21314	177956	5719880
2^{216}	9453	39218	329092	10601480
2^{250}	11170	46460	390360	12582320
2^{305}	14030	58560	492880	15899040
2^{372}	17514	73300	617768	19939408
2^{486}	23769	99906	843780	27259656

As a concrete example, the strategy for computing a 2^{602}-isogeny in dimension 2 involves 3274 doublings and 7108 images (plus 26 gluing images).

17.1. **Removing inversions.** It is also much easier to analyze the complexity where we get rid of all inversions. We treat the case $g = 2$ for simplicity.

We compute $h(P_1) = (Ax_1 : Bx_1 : Cz_1 : Dz_1)$ and $h(P_2) = (Ax_2 : Bz_2 : Cx_2 : Dz_2)$ in $2.2^g S = 8$, since $g = 2$. From this data we want $(1/A : 1/B : 1/C : 1/D)$ projectively for images, and also $(1/a_2 : 1/b_2 : 1/c_2 : 1/d_2)$ for doublings.

Batching inversions, we can compute $B/A, C/A, D/C$ in $1I + 6M + 3M$, except we don't want to actually compute the inversion so we have $(\kappa B/A, \kappa C/A, \kappa D/C)$ for some known factor λ. We then get $(\kappa^2, \kappa^2 B/A, \kappa^2 C/A, \kappa^2 D/C)$ in $1S + 3M$. We compute the inverse of these coordinates in $1I + 9M$ (except we don't actually compute the inverse), and likewise we have $(a_2 : b_2 : c_2 : d_2)$ through a Hadamard transform of $(A : B : C : D)$ and then compute the inverses in $1I + 9M$. The total cost for the codomain is than $8S + 9M + (1S + 3M) + 9M + 9M = 30M + 9S \geq 39M$. Comparing with Section 16 we see that the codomain computation is more than twice as fast, as was expected (since the normalisation process took half the time). Here, images and doublings costs $2^g S + 2^g M \leq 2.2^g M$ and $2.(2^g S + 2^g M) \leq 4.2^g M$ respectively because the coordinates of our theta null points are no longer normalised.

Depending on the number of doubling and isogeny images we need, it might make sense (especially at the beginning of the isogeny chain) to bach one inversion to gain the $1M$ (resp. $2M$) by doublings and images.

We obtain the following number of arithmetic operations, when we get rid of all inversions:

2^n	$g = 2$
2^{128}	25840
2^{216}	47320
2^{250}	55990
2^{305}	70455
2^{372}	88076
2^{486}	119802

References

[BRS23] R. Barbulescu, D. Robert, and N. Sarkis. "Models of Kummer lines and Galois representations". June 2023. In preparation. (Cit. on pp. 9, 30).

[BMP23] A. Basso, L. Maino, and G. Pope. "FESTA: Fast Encryption from Supersingular Torsion Attacks". In: *Cryptology ePrint Archive* (2023) (cit. on p. 4).

[CD21] W. Castryck and T. Decru. "Multiradical isogenies". In: *Cryptology ePrint Archive* (2021) (cit. on p. 36).

[CD23] W. Castryck and T. Decru. "An efficient key recovery attack on SIDH". In: Springer-Verlag (Eurocrypt 2023), Apr. 2023 (cit. on p. 4).

[CR15] R. Cosset and D. Robert. "An algorithm for computing (ℓ,ℓ)-isogenies in polynomial time on Jacobians of hyperelliptic curves of genus 2". In: *Mathematics of Computation* 84.294 (Nov. 2015), pp. 1953–1975. DOI: 10.1090/S0025-5718-2014-02899-8. URL: http://www.normalesup.org/~robert/pro/publications/articles/niveau.pdf. HAL: hal-00578991, eprint: 2011/143. (Cit. on pp. 4, 5, 36).

[CH17] C. Costello and H. 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 23*. Springer. 2017, pp. 303–329 (cit. on p. 21).

[CLN16] C. Costello, P. Longa, and M. Naehrig. "Efficient algorithms for supersingular isogeny Diffie-Hellman". In: *Advances in Cryptology–CRYPTO 2016: 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I 36*. Springer. 2016, pp. 572–601 (cit. on p. 21).

[DLRW23] P. Dartois, A. Leroux, D. Robert, and B. Wesolowski. "SQISignHD: New Dimensions in Cryptography". Accepted for publication at Eurocrypt 2024. Mar. 2023. URL: http://www.normalesup.org/~robert/pro/publications/articles/SQISignHD.pdf. eprint: 2023/436, HAL: hal-04056062v1. (Cit. on pp. 3–5, 7, 8, 11, 12, 15, 23).

[DMPR23a] P. Dartois, L. Maino, G. Pope, and D. Robert. "An Algorithmic Approach to (2, 2)-isogenies in the Theta Model and Applications to Isogeny-based Cryptography". Nov. 2023. URL: http://www.normalesup.org/~robert/pro/publications/articles/_2_2__isogenies_in_the_theta_model.pdf. eprint: 2023/1747. (Cit. on p. 4).

[DMPR23b] P. Dartois, L. Maino, G. Pope, and D. Robert. *ThetaIsogenies*. Fast computations of isogenies in dimension two. Nov. 2023. URL: https://github.com/ThetaIsogenies/two-isogenies (cit. on p. 4).

[DJP14] L. De Feo, D. Jao, and J. Plût. "Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies". In: *Journal of Mathematical Cryptology* 8.3 (2014), pp. 209–247 (cit. on pp. 4, 16).

[FLR11] J.-C. Faugère, D. Lubicz, and D. Robert. "Computing modular correspondences for abelian varieties". In: *Journal of Algebra* 343.1 (Oct. 2011), pp. 248–277. DOI: 10.1016/j.jalgebra.2011.06.031. arXiv: 0910.4668 [cs.SC]. URL: http://www.normalesup.org/~robert/pro/publications/articles/modular.pdf. HAL: hal-00426338. (Cit. on p. 36).

[LR10] D. Lubicz and D. Robert. "Efficient pairing computation with theta functions". In: ed. by G. Hanrot, F. Morain, and E. Thomé. Vol. 6197. Lecture Notes in Comput. Sci. 9th International Symposium, Nancy, France, ANTS-IX, July 19-23, 2010, Proceedings. Springer–Verlag, July 2010. DOI: 10.1007/978-3-642-14518-6_21. URL: http://www.normalesup.org/~robert/pro/publications/articles/pairings.pdf. Slides: 2010-07-ANTS-Nancy.pdf (30min, International Algorithmic Number Theory Symposium (ANTS-IX), July 2010, Nancy), HAL: hal-00528944. (Cit. on p. 8).

[LR12] D. Lubicz and D. Robert. "Computing isogenies between abelian varieties". In: *Compositio Mathematica* 148.5 (Sept. 2012), pp. 1483–1515. DOI: 10.1112/S0010437X12000243. arXiv: 1001.2016 [math.AG]. URL: http://www.normalesup.org/~robert/pro/publications/articles/isogenies.pdf. HAL: hal-00446062. (Cit. on pp. 4, 5, 36).

[LR15a] D. Lubicz and D. Robert. "A generalisation of Miller's algorithm and applications to pairing computations on abelian varieties". In: *Journal of Symbolic Computation* 67 (Mar. 2015), pp. 68–92. DOI: 10.1016/j.jsc.2014.08.001. URL: http://www.normalesup.org/~robert/pro/publications/articles/optimal.pdf. HAL: hal-00806923, eprint: 2013/192. (Cit. on p. 8).

[LR15b] D. Lubicz and D. Robert. "Computing separable isogenies in quasi-optimal time". In: *LMS Journal of Computation and Mathematics* 18 (1 Feb. 2015), pp. 198–216. DOI: 10.1112/S146115701400045X. arXiv: 1402.3628. URL: http://www.normalesup.org/~robert/pro/publications/articles/rational.pdf. HAL: hal-00954895. (Cit. on pp. 4, 5).

[LR16] D. Lubicz and D. Robert. "Arithmetic on Abelian and Kummer Varieties". In: *Finite Fields and Their Applications* 39 (May 2016), pp. 130–158. DOI: 10.1016/j.ffa.2016.01.009. URL: http://www.normalesup.org/~robert/pro/publications/articles/arithmetic.pdf. HAL: hal-01057467, eprint: 2014/493. (Cit. on p. 8).

[LR22a] D. Lubicz and D. Robert. "Fast change of level and applications to isogenies". In: *Research in Number Theory (ANTS XV Conference)* 9.1 (Dec. 2022). DOI: 10.1007/s40993-022-00407-9. URL: http://www.normalesup.org/~robert/pro/publications/articles/change_level.pdf. HAL: hal-03738315. (Cit. on pp. 4, 5, 8, 36).

[LR22b] D. Lubicz and D. Robert. "Multiradical isogenies in the theta model". Sept. 2022. In preparation. (Cit. on p. 36).

[MMPPW23] L. Maino, C. Martindale, L. Panny, G. Pope, and B. Wesolowski. "A Direct Key Recovery Attack on SIDH". In: Springer-Verlag (Eurocrypt 2023), 2023 (cit. on p. 4).

[Mum66] D. Mumford. "On the equations defining abelian varieties. I". In: *Invent. Math.* 1 (1966), pp. 287–354 (cit. on p. 5).

[Ren18] J. Renes. "Computing isogenies between Montgomery curves using the action of (0, 0)". In: *Post-Quantum Cryptography: 9th International Conference, PQCrypto 2018, Fort Lauderdale, FL, USA, April 9-11, 2018, Proceedings*. Springer. 2018, pp. 229–247 (cit. on p. 21).

[Rob10] D. Robert. "Theta functions and cryptographic applications". PhD thesis. Université Henri-Poincarré, Nancy 1, France, July 2010. URL: http://www.normalesup.org/~robert/pro/publications/academic/phd.pdf. Slides: 2010-07-Phd-Nancy.pdf (1h, Nancy), TEL: tel-00528942. (Cit. on pp. 4, 5, 8, 9, 11, 12, 32, 34).

[Rob21] D. Robert. "Efficient algorithms for abelian varieties and their moduli spaces". HDR thesis. Université Bordeaux, June 2021. URL: http://www.normalesup.org/~robert/pro/publications/academic/hdr.pdf. Slides: 2021-06-HDR-Bordeaux.pdf (1h, Bordeaux). (Cit. on pp. 4, 5, 8, 9).

[Rob23a] D. Robert. "Breaking SIDH in polynomial time". Apr. 2023. URL: http://www.normalesup.org/~robert/pro/publications/articles/breaking_sidh.pdf. eprint: 2022/1038, HAL: hal-03943959, Slides: 2023-04-Eurocrypt.pdf (15 min, Eurocrypt 2023, April 2023, Lyon, France). (Cit. on p. 4).

[Rob23b] D. Robert. "Improving the arithmetic of Kummer lines". Aug. 2023. URL: http://www.normalesup.org/~robert/pro/publications/notes/2023-11-kummer_lines.pdf (cit. on p. 21).

[RS24] D. Robert and N. Sarkis. "Computing 2-isogenies between Kummer lines". Jan. 2024. URL: http://www.normalesup.org/~robert/pro/publications/articles/kummer_isogenies.pdf. eprint: 2024/037. (Cit. on p. 21).

Appendix A. Conversion formula between the theta model and the Montgomery model in dimension 1

These formula are extracted from [BRS23].

A.1. **Theta and Montgomery.** Let E/k be an elliptic curve, and $(a : b) = (\theta_0(0_E), \theta_1(0_E))$ be its theta null point. We give formula to convert the theta points $(\theta_0(P) : \theta_1(P))$ into the Montgomery coordinates $(x(P) : z(P))$.

When the theta null point is rational, the elliptic curve E admits both a rational Montgomery model and a rational Legendre model. They are given by

$$y^2 = x(x - \alpha)(x - 1/\alpha) = x(x^2 + Ax + 1)$$

and (up to a quadratic twist, which is harmless because we work on the Kummer line anyway) by

$$y^2 = x(x - 1)(x - \lambda).$$

These constants are determined as follows: let $(A : B)$ be the dual coordinates of the canonical 2-isogenous curve (we will only need their square). We have

$$A^2 = a^2 + b^2, B^2 = a^2 - b^2, \tag{4}$$

$$\alpha = A^2/B^2 = (a^2 + b^2)/(a^2 - b^2), \tag{5}$$

$$\lambda = \alpha^2 = A^4/B^4 = (a^2 + b^2)^2/(a^2 - b^2)^2, \tag{6}$$

$$\mathcal{A} = -(\alpha + 1/\alpha) = -(\alpha^2 + 1)/\alpha = -(A^4 + B^4)/(A^2B^2) = -2(a^4 + b^4)/(a^4 - b^4), \tag{7}$$

$$(\mathcal{A} + 2)/4 = -b^4/(a^4 - b^4). \tag{8}$$

Conversely, from $\mathcal{A}$, we can recover $(a : b)$ via

$$\alpha + 1/\alpha = -\mathcal{A}, \tag{9}$$

$$A^2/B^2 = \alpha, \tag{10}$$

$$a^2 = A^2 + B^2, b^2 = A^2 - B^2, (a^2 : b^2) = (\alpha + 1 : \alpha - 1). \tag{11}$$

We note that if $(a : b)$ is a solution, then $(a : \zeta b)$ also with $\zeta \in \mu_4$, these correspond to different theta structures.

With these constants defined, we can now explain how to convert the points. If $P = (x : z)$ in Montgomery coordinates, then

$$(\theta_0(P) : \theta_1(P)) = (a(x - z) : b(x + z)). \tag{12}$$

Conversely, if $P = (\theta_0 : \theta_1)$, then in Montgomery coordinates

$$(x(P) : z(P)) = (a\theta_1 + b\theta_0 : a\theta_1 - b\theta_0). \tag{13}$$

On the theta model $0_E = (a : b)$, we have a canonical basis of the 2-torsion given by $T_1 = (a : -b)$ and $T_2 = (b : a)$. We have a canonical basis of the 4-torsion given by $T_1' = (1 : 0)$ above T_1 and $T_2' = (1 : 1)$ above T_2. The map above sends T_1 to $(0 : 1)$ in the Montgomery model, T_1' to $(1 : 1)$, T_2 to $(A^2 : B^2)$, T_2' to $(a + b : a - b)$.

So conversely, given a Montgomery curve, the canonical point $T' = (1 : 1)$ of 4-torsion above the 2-torsion point $T = (0 : 1)$ and a second point $T'' = (r : s)$ above another point of 2-torsion, then the theta null point $(a : b)$ induced by the basis (T', T'') of the 4-torsion is given by $(r + s : r - s)$.

For the case of a general elliptic curve E with a basis (T', T'') of the 4-torsion, we first convert E to a Montgomery model by sending T' to $(1 : 1)$ and $T = 2T'$ to $(0 : 1)$, the map is then $x \mapsto (x - x(T))/(x(T') - x(T))$. Then we apply the above formula to the image of T''.

A.2. **The alternative Montgomery model.** When we have a theta model, we can also introduce the dual theta coordinates

$$(\theta_0' : \theta_1') = (\theta_0 + \theta_1 : \theta_0 - \theta_1),$$

in particular the dual theta null point is given by $(a' : b') = (a + b : a - b)$. We can construct another Montgomery model by replacing in the above formula $(a, b, \theta_0, \theta_1)$ by $(a', b', \theta_0', \theta_1')$.

Plugging in this different model the equations expressing $(a', b', \theta'_0, \theta'_1)$ in terms of $(a, b, \theta_0, \theta_1)$, we obtain alternative formulas:

$$A'^2 = a'^2 + b'^2 = 2(a^2 + b^2), B'^2 = a'^2 - b'^2 = 4ab, \tag{14}$$

$$\alpha' = A'^2/B'^2 = (a^2 + b^2)/(2ab), \lambda' = \alpha'^2, \tag{15}$$

$$\mathcal{A}' = -(\alpha' + 1/\alpha') = -(a^4 + 6a^2b^2 + b^4)/(2(a^3b + ab^3)), \tag{16}$$

$$P = (x : z) \mapsto (\theta_0(P), \theta_1(P)) = (ax - bz : bx - az), \tag{17}$$

$$(\theta_0, \theta_1) \mapsto (x(P) : z(P)) = (a\theta_0 - b\theta_1 : b\theta_0 - a\theta_1). \tag{18}$$

Appendix B. The algebraic theta transformation formula

We briefly describe the algebraic theta transformation formula in level n, see [Rob10] for more details.

Assume that we have a symmetric theta structure of level n, induced by a symplectic basis $(e'_1, \dots, e'_g, f'_1, \dots, f'_g)$ of $A[2n]$. Let $M \in \mathrm{Sp}_{2g}(\mathbb{Z})$, this induces a symplectic matrix of $\mathbb{Z}/n\mathbb{Z}$ hence a symplectic change of variable on our basis above. The new symplectic basis will give a new symmetric theta structure, hence a linear change of variable on our theta functions. We now describe this action.

The group $\mathrm{Sp}_{2g}(\mathbb{Z})$ is generated by these three types of matrices:

(1) The matrix $\mathcal{S} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$. This matrix acts by the Hadamard transform H.

(2) The matrix $M = \begin{pmatrix} A & 0 \\ 0 & A^{-T} \end{pmatrix}$ where A is in $\mathrm{Gl}_g(\mathbb{Z})$. This matrix acts by $\theta_i \mapsto \theta_{A.i}$, where the action is the natural action of A on $(\mathbb{Z}/n\mathbb{Z})^g$ induced by the action of A on $\mathbb{Z}^g$.

(3) The matrix $M = \begin{pmatrix} 1 & C \\ 0 & 1 \end{pmatrix}$ where C is symmetric. Fix ζ a primitive $2n$-root of unity induced by a symplectic basis of the $2n$-torsion inducing our theta strucutre. This matrix acts by $\theta_i \mapsto \zeta^{i^T C i}\theta_i$. For instance if C is diagonal with the only non zero entry being a one at position (j, j), the action is $\theta_x \mapsto \zeta^{x_j^2}\theta_x$. If C is diagonal with only two non zero entries at position $(i, j), (j, i)$, the action is $\theta_x \mapsto \zeta^{2x_i x_j}\theta_x$.

Example B.1. In dimension 1, $\Gamma/\Gamma(2, 4)$ is of cardinal $6 * 4 = 24$, the modular action induces all possible permutation on the four points of ramification of $E \to \mathbb{P}^1$.

Example B.2. In dimension 2, $\Gamma/\Gamma(2, 4)$ is of cardinal $720 * 2^4$. If the abelian surface is a product of two elliptic curves, the subgroup preserving a product theta structure is of cardinal $2 * 24 * 24$ so is of index 10. There are ten even theta constants of level $(2, 2)$, an abelian surface is a product theta if and only if the even theta constant $\theta[11; 11](0) = 0$. The index 10 corresponds to sending this null theta constant to one of the other 10 even theta.

B.1. Directly computing theta constants.

The original proposal of these notes suggested to compute the theta constants in level 2 by going through the product theta structure (via the dimension 1 conversion of Appendix A) followed by a symplectic transform.

However, Sage's linear algebra is quite slow, so the current implementation directly computes the theta constants from a symplectic basis on the elliptic product.

An advantage of this approach is as follow: going to the direct product theta structure involve starting with a tuple (P_1, P_2) in Montgomery coordinate, applying a linear transform (in dimension 1) on the coordinates of P_i to obtain theta coordinates, take the Segre

embedding to get the product theta structure coordinates, and apply a linear transform again (in dimension 2) to get the theta coordinates compatible with our kernel. With the direct approach we take the Segre embedding on the Montgomery coordinate and directly apply a dimension 2 base change; this save the 2 dimension 1 linear base change.

We briefly explain how this works: on a dimension 1 theta model $(a : b)$, the point $T_2 = (-a : b)$ as for symmetric lifts in the theta group the linear transformation $(X, Z) \mapsto (X, -Z)$ (associated to the 4-torsion point $(1 : 0)$) and $(X, Z) \mapsto (-X, Z)$ (associated to the 4-torsion point $(0 : 1)$). And the point $T_1 = (b : a)$ as for symmetric lifts the linear transformation $(X, Z) \mapsto (Z, X)$ (associated to the 4-torsion point $(1 : 1)$) and the linear transformation $(X, Z) \mapsto (-Z, -X)$ (associated to the 4-torsion point $(-1 : 1)$).

From the conversion maps Appendix A, we see that on a Montgomery curve, the 4-torsion point $T = (1 : 1)$ above $(0 : 1)$ is associated to the linear map $g_T : (X, Z) \mapsto (-Z, -X)$, while the point $(-1 : 1)$ is associated to $(X, Z) \mapsto (Z, X)$.

For a general elliptic curve, if $T = (x, y, z)$ is a point of 4-torsion and $2T = (u, v, w)$, we can map T to the Montgomery point $(1 : 1)$ via the linear transformation (in the Kummer line): $(X : Z) \mapsto (X', Z') = (zwX - zuZ : (xw - zu)Z)$. It follows that in (X, Z) coordinates, the action of g_T is given by

$$M^T U M^{-T} = \begin{pmatrix} uz/(wx - uz) & zw/(wx - uz) \\ (-wx^2 + 2uzx)/(-zwx + uz^2) & -uz/(wx - uz) \end{pmatrix},$$

with $M = \begin{pmatrix} wz & -zu \\ 0 & xw - uz \end{pmatrix}$, $U = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$.

Given a symplectic decomposition $A[2] = K_1 \oplus K_2$ and a decomposition $A[4] = K_1' \oplus K_2'$ above it, and a section $s \in \Gamma(L)$ (L of level 2), we can construct a basis of level 2 theta functions by taking the trace (provided it is non zero) θ_0 of s under the level 2 elements induced by the linear transformation $g_{T'}$, $T' \in K_2'$ above each $T \in K_2$. Then for $i \in K_1$, $i' \in K_1'$ above i, we let $\theta_i = g_{i'} \cdot \theta_0$.

As an example, on a Montgomery curve we have $T_2 = (-1 : 1)$ which acts by $g_2 \cdot (X, Z) = (-Z, -X)$. Taking the trace of X under this action we get: $\theta_0 = \mathrm{id} \cdot X + g_1 \cdot X = X - Z$.

Let $T_1 = (a+b : a-b)$ be another point of 4-torsion; its double is then $(a^2+b^2 : a^2-b^2)$. So with $x = a + b$,$z = a - b$,$u = a^2 + b^2$,$w = a^2 - b^2$, we compute $\theta_1 = g_1 \cdot \theta_0 = g_1 \cdot (X - Z) = (uz - wxx/z + 2ux)/(wx - uz)X + z(w + u)/(wx - uz)Z = b/aX + b/aZ$. Hence we recover exactly the base change $(X, Z) \mapsto (a(X - Z) : b(X + Z))$ of Appendix A from Montgomery to theta.

We can use the same strategy to compute the theta null point associated to a symplectic basis of the 4-torsion on a product of elliptic curve. If $T = (T_1, T_2) \in E_1 \times E_2$ is a point of 4-torsion, the associated element g_T is given by $g_T = g_{T_1} \otimes g_{T_2}$. Then we can (for instance) take θ_0 as the trace of $X_1 \otimes X_2$ under $\widetilde{K}_2$, with $(g_{T_1} \otimes g_{T_2}) \cdot (X_1 \otimes X_2) = (g_{T_1} \cdot X_1) \otimes (g_{T_2} \cdot X_2)$.

B.2. **The choice of signs.** We can use this action to explore our choice of sign. Fix $(e_1', \dots, e_g', f_1', \dots, f_g')$ a symplectic basis of $A[4]$ inducing our symmetric theta structure, and let $K = \langle f_1, \dots, f_g \rangle$ where $f_i = 2f_i'$ our kernel, and let $f : A \to B = A/K$. The image $f(e_i')$ gives an isotropic subgroup $B_1[4]$ of $B[4]$, while $f(f_i')$ gives $B_2[2]$ such that we have a symplectic decomposition $B[2] = B_1[2] \oplus B_2[2]$. The choice of sign in our isogenous theta constant corresponds to fixing a symplectic basis of $B[4]$ compatible with the $f(e_i')$,$f(f_i)$.

These choice of signs corresponds to the action of the matrix $M = \begin{pmatrix} 1 & C \\ 0 & 1 \end{pmatrix}$ on B. We note that there are two kind of action: the one where C leaves $B[2]$ invariant, this corresponds on A to leaving the f_i' invariant and changing the points of 8-torsion above them. An example

is given by C which is diagonal with entries equal to 0 modulo 2. The second type changes $B[2]$, hence changes the f'_i (but in a way that is still compatible with our theta structure on A).

This shows that we have $g(g+1)/2$ choice of sign possible on B (because the matrix C has to be symmetric), but that if we fix the 4-torsion f'_1, f'_2 on A we now only have g choice of signs. Algebraically these can be determined as follow: normalize the f'_i, because f'_i is of order 4 this still leaves a choice of sign for each f'_i (specifically, we have an equation $\lambda_i^4 = C_i$, but the duplication formula only involve the λ_i^2). This is enough to compute the isogenous theta null point by Section 11.

In fact this is also enough to also normalize all of $K[4]$: normalize $f'_i + f'_j$ via $\widetilde{f'_i + f'_j} + \widetilde{f'_j} = \widetilde{f'_i} + \tilde{f}_j$, this involve an equation $\lambda_{ij}^2 = C_{ij}$ so no choice of sign since the duplication formula only involve the λ_{ij}^2. Next use the differential additions and three way additions to normalize any remaining point. See [Rob10] for more details.

Example B.3. When $g = 1$, our point of 4-torsion is $T' = (1 : 0)$, which is normalized to $(\lambda, 0)$ where $\lambda^4 = A^2B^2$. Since we know $A^2 = a^2 + b^2, B^2 = a^2 - b^2$, fixing $\lambda^2 = AB$ is enough to fix $(A : B)$; changing to $\lambda^2 = -AB$ gives $(A : -B)$, and this correspond to normalising T' by another point of 8-torsion above it.

When $g = 2$, let's say that our point of 8-torsion determined the coefficients $(A : B : C : D)$. Our points of four torsion (suitably normalised by the 8-torsion) is then f'_1 which determines (AB, CD, AB, CD) and f'_2 which determines (AC, BD, AC, BD). Changing f'_1 by $f'_1 + e_2$ and f'_2 by $f'_2 + e_1$ will give instead the coefficients $(AB, -CD, AB, -CD)$ and $(AC, -BD, AC, -BD)$, hence corresponds to changing the sign of D. Keeping f'_1 and f'_2 but changing the points of 8-torsion above f'_1, hence changing their normalisation, then f'_1 will now give $(-AB, -CD, -AB, -CD)$, hence this changes the sign of B and D. Similarly changing the point of 8-torsion above f'_2 will change the sign of C and D. We do recover that we have 2 possible choice of signs when the f'_1, f'_2 are fixed, and one more when we change them (while staying compatible with the theta structure).

By the way, we remark that if we don't normalize f'_1, f'_2, we recover $\lambda_1 AB, \lambda_1 CD, \lambda_2 AC, \lambda_2 BD$ for some unknown projective factors λ_1, λ_2. Since we also know A^2, B^2, C^2, D^2, it is easy to find equations for λ_1^2, λ_2^2. By the above discussion, all 4 solutions of these two equations determine a valid isogenous (dual) theta null point; but they require to take two square roots.

The advantage of requiring points of 8-torsion is to dispense with these square roots. However, for a 2^n-isogeny, this requires to start with points of 2^{n+2}-torsion above our kernel K. If we only have K, we could switch to the square root method for the second to last (which requires 2 square roots because we have points of 4-torsion), and last (which requires 3 square roots because we now only have the 2-torsion) isogenies. This slow down the last two codomain computations, however this does not change the images.

Example B.4. When $g = 2$, it was remarked by Giacomo Pope that we don't need to start our isogeny chain with isotropic 2^{n+2}-torsion points P''_1, P''_2 above the kernel P_1, P_2, we just need non necessarily isotropic points.

The first $n-2$ steps are the same, we just need to explain why the formula still work at the last two steps.

At the penultimate step, assume that our 8-torsion T''_1, T''_2 is correct and that $f(T''_1) = (x : x : y : y)$, and $f(T''_2) = (z : t : z : t)$ in θ' coordinate. If we change our points by $T \in K_2$ this changes nothing because K_2 is our kernel so the images are the same. If $T \in K_1$, T acts by a shift in θ coordinates, so by a sign in θ'-coordinate. So if T''_2 is wrong we could

have $f(T''_2) = (z : t : -z : -t)$. If we look at our code, we see that this changes by a sign one of the constant $(A : B : C : D)$ we compute.

So when the 4-torsion is correct, but the 8 torsion is wrong, then in the codomain the 4-torsion is wrong, rather than getting $(x : x : y : y)$ say above $(B : A : D : C)$, we get $(x : -x : y : -y)$. So instead of getting (A, B, C, D), we get $-B, -D$, but $(A : -B : C : -D)$ is still a valid theta null point. But compared to our true image map, our image map has sign flips. In particular, the 4-torsion point which was sent to $(B : A : D : C)$ before, is now sent to $(B : -A : D : -C)$. But compared our new theta null point $(A : -B : C : -D)$, this is still the correct 2-torsion point for the next isogeny!

We can do a similar reasoning for the last isogeny. Here even the 4-torsion is wrong, so our 2-torsion on the codomain is wrong: say rather than getting $(B : A : D : C)$ we get $(B : -A : D : -C)$. Then it follows that our 8-torsion point is sent to a 4-torsion point above this 2-torsion, which means it is of the form (x, ix, y, iy) or $(x, -ix, y, -iy)$. If we use this point, this will change B to iB say (and maybe for D too), but this kind of change also comes from a symplectic automorphism. And since we don't take any more kernel, we don't care what the images of our 4-torsion points are anyway afterwards.

Appendix C. Other applications of the duplication formula

By now the reader should be convinced that the duplication formula for theta functions allows for very fast 2-isogeny formula. The Sage implementation (due to Pierrick Dartois, Sabrina Kunzweiler, Luciano Maino, Giacomo Pope and myself) shows a nice speed up compared to Richelot formula, this will be detailed in a follow up work.

One can wonder if theta coordinates can be used for other applications. The following use case was suggested by Sabrina Kunzweiler in dimension 2: look at CGL like hash function in dimension 2 in the theta model, by computing chuncks of 2^n-isogenies.

Altough the formula from these notes can be used, there are two problems remaining:

- Compute the symplectic change of basis to make the kernels K in a way such that $K[4]$ is canonical. A method is probably to compute a basis of $A[4]$ compatible with our theta structure (unfortunately this involves square roots), as described in Section 16.6, complete $K[4]$ into a symplectic basis, compute the symplectic change of basis, eg using Weil pairings (but since we are in level 2 this involves more square roots), and apply the theta transformation formula. If we start with a Jacobian, and we compute the theta constants through Thomaes formula, a better method would be to use formula due to Sabrina which gives the correct square roots to take in Thomae's formula according to a fixed symplectic basis of $\mathrm{Jac}\, C[4]$.
- Once K is in suitable form, and more generally we have e'_1, e'_2, f'_1, f'_2 a symplectic basis of $A[2^n]$, such that the induced symplectic basis e_1, e_2, f_1, f_2 a symplectic basis of $A[4]$ is compatible with the theta structure and $K = \langle f'_1, f'_2 \rangle$ (so that $\langle f_1, f_2 \rangle = K[4]$ and K is compatible with our theta structure); we can compute the isogeny $f : A \to B$ and the images $f(e'_1), f(e'_2)$.

  We now need to regenerate the 2^n-torsion of B by computing a symplectic basis $f(e'_1), f(e'_2), g'_1, g'_2$ and take a kernel K' whose intersection with $\langle f(e'_1), f(e'_2) \rangle$ is trivial (so the next isogeny has no (partial) backtracking.

  Since we are in level 2 however it is not clear how to best do this step. Sample random points, multiply by the cofactor, and do some Weil pairing computations (which as mentioned involve square roots since we are in level 2)? Go back to a Jacobian representation to compute the 2^n-torsion and switch back to theta afterwards?

  We leave the best method as an open problem.

What is much easier though is to only do chuncks of 2-isogenies and use multiradical 2-isogeny formula in the spirit of [CD21] to regenerate the 2-torsion at each step (in a non backtracking way). Remember that in dimension g, multiradical formula will involve $g(g+1)/2$ square roots.

Using the duplication formula, 2-isogeny multiradical formula are particularly simple in the level 2 theta model in dimension 1 and 2:

- In dimension 1 start with the theta null point $(a : b)$, apply the square operator S to get $(a^2 : b^2)$, the Hadamard operator $H(x : y) = (x + y : x - y)$ to get $(A^2 : B^2) = (a^2 + b^2 : a^2 - b^2)$. Take an arbitrary square root of B^2/A^2. To prevent an inversion, a solution is to instead take an arbitrary square root AB (depending on the current bit of the message we want to hash) of A^2B^2, which give the projective dual isogeneous theta null point $(A^2 : AB)$. Apply the Hadamard operator H again to get $(a_2 : b_2)$, this is our isogeneous theta null point. Iterate for each bit of our message. The whole formula cost one square root, $2S + 1M + 4a$.
- In dimension 2, the same formula hold: start with the theta null point $(a : b : c : d)$, apply S to get $(a^2 : b^2 : c^2 : d^2)$, then $H(x : y : z : t) = (x + y + z + t, x - y + z - t, x + y - z - t, x - y - z + t)$ to get $(A^2 : B^2 : C^2 : D^2)$, take arbitrary square roots (depending on our bits) AB, AC, AD of A^2B^2, A^2C^2, A^2D^2 to get $(A^2 : AB : AC : AD)$, and apply H again to get $(a_2 : b_2 : c_2 : d_2)$ for a total cost of three square roots and $4S + 3m + 8a$.

It would be interesting to compare these methods with the usual methods:

- In dimension 1 using the modular polynomial ϕ_2, removing the linear factor coming from the preceding isogeny and solving a degree 2 equation
- In dimension 2 using Richelot formula, factorizing 3 degree 2 polynomials at each step.

We also leave that for future work!

An interesting open problem is to generalize this approach to higher dimension. From the theta transformation formula, one can see that we can only take $g(g+1)/2$ arbitrary square roots (the ones coming from $e_i, e_i + e_j$ where e_i is a basis of $(\mathbb{Z}/2\mathbb{Z})^g$), once these are taken the rest are fixed. But I don't know how to most efficiently determine these remaining choices (apart from a rather expensive Grobner basis computation), unless we already have some information on the 4-torsion on the domain. When $g = 1, 2$, all choices are possible, so this problem goes away.

Another interesting direction is to extend these 2-radical formulas to 4-radical and 8-radical formula. Using the generic isogeny algorithm [LR12; CR15; LR22a] combined with [FLR11], I have a generic multiradical isogeny formula in any dimension in the theta model [LR22b]. But we have just seen that 2^n-isogenies in the theta model can be made much faster than the generic isogeny computation, so it's probably better to find direct radical isogeny formula for $\ell = 4, 8$.

INRIA Bordeaux–Sud-Ouest, 200 avenue de la Vieille Tour, 33405 Talence Cedex FRANCE
Email address: damien.robert@inria.fr
URL: http://www.normalesup.org/~robert/

Institut de Mathématiques de Bordeaux, 351 cours de la liberation, 33405 Talence cedex FRANCE

8 IMPROVING THE ARITHMETIC OF KUMMER LINES (2023-11)

These notes are available at http://www.normalesup.org/~robert/pro/publications/notes/2023-11-kummer_lines.pdf.

These notes cover results obtained between the end of 2022 and the beginning 2023 on improving the arithmetic of Kummer lines (and Kummer surfaces). Part of these results were obtained in collaboration with Barbulescu and Sarkis. A small part of these results are now published in [RS24]. The arithmetic of biextension is detailed in more details in the slides http://www.normalesup.org/~robert/pro/publications/slides/2023-10-Leuven.pdf of my presentation "Arithmetic and pairings on Kummer lines" for the Leuven isogeny days 4. The code of the algorithms described in these notes is available at [Rob23].

Improving the arithmetic of Kummer lines

DAMIEN ROBERT

Abstract. We explain some improvements to the arithmetic of Kummer lines: doublings, differential additions, scalar multiplications, pairings, isogenies.

Contents

Date: November 2023.

1. Introduction

This is a summery of results that will be presented in a series of articles on the arithmetic of Kummer lines.

- In [BRS23], we focus on the general theory of models of Kummer lines, the conversions between them, and the arithmetic properties of their 2-torsion points (with the relationship between the ramification, the 2-Tate pairing, the 2-theta group and their Galois representation).
- In [RS24], we study isogenies between Kummer lines, and in particular we focus on 2-isogenies. We use the action of the theta group $G(2(0_E))$ rather than Vélu's formula to compute invariant sections, and the fact that the Kummer model is determined by its ramification, to find new and old formulas. This allows us to give a general framework to find equations for 2-isogenies and doublings. We also develop an hybrid arithmetic, combining the best of the (twisted) theta and Montgomery models.
- In [Rob22], we extend the work of [RS24] from doublings to differential additions on a Kummer model (the formula crucially depend on the arithmetic property of the 2-torsion alluded to above). Notably, we explain how to find differential additions formulae which factor through a 2-isogeny. As an application we develop a novel time/memory trade off for the Montgomery ladder.
- In [Rob23c] we develop the arithmetic of the biextension associated to the divisor $2(0_E)$ on some Kummer models. We extend this to the effective computation of the cubical torsor structure. We derive from this efficient pairing formulae.
- In [Rob23f], we use the formula from [Rob23c] to revisit the "Projective coordinates leak" paper [NSS04]. We show that leaking the projectives coordinate in the Montgomery ladder yields a subexponential time recovery of the full secret key (by reduction to the DLP over the base field). The previous attacks only recovered a few bits by leak.

A proof of concept implementation of these algorithms is available in [Rob23e].

2. Summary

In isogeny based cryptography, it is standard to work with the Montgomery model of a Kummer line. In the case where we have an extra point of 2-torsion T_2 along with the standard point of 2-torsion $T_1 = (0 : 1)$ (as happens for supersingular curves), we can use T_2 to speed up the arithmetic.

2.1. **Hybrid arithmetic.** (This is joint work with Nicolas Sarkis):

2.1.1. *Hybrid arithmetic for scalar multiplication.* In the Montgomery ladder for computing $m.P$, we use one doubling and one mixed differential addition by step. In the Montgomery model, doubling is $2M + 2S + 1m_0$ while a mdiffAdd (where we assume our base point $P = (X_P : 1)$ is normalised) is $3M + 2S$, so a ladder step is $5M + 4S + 1m_0$. Here m_0 denotes a multiplication by a curve constant (typically the coefficient A of the Montgomery curve, or rather $(A + 2)/4$). If our starting point $P = (X_P : Z_P)$ is not normalised, we need to add $1M$ by bit to the ladder cost.

When T_2 is rational, we can also use a twisted theta model, where doubling is $4S + 2m_0$, and mdiffAdd is $3M+2S+1m_0$, so a ladder step is $3M+6S+3m_0$. (There is a $1M-1S-1m_0$ tradeoff where a ladder step is $4M + 5S + 2m_0$.)

The two models differ by the translation by T_2 (the doubling $P \mapsto 2.P$ in twisted theta can be interpreted as $P \mapsto 2.P + T_2$ in the Montgomery model and conversely), we can exploit that to combine the best of both worlds: using an hybrid arithmetic where doubling is $4S + 2m_0$ and mdiffAdd is $3M + 2S$. Keeping track of the translation by T_2 we then have a hybrid ladder which cost $3M + 6S + 2m_0$.

2.1.2. *Hybrid arithmetic for 2^n-isogenies.* In the Montgomery model, a 2-isogeny codomain costs $2S$, an image costs $4M$, and doubling cost $2S+2M+2m_0$ (because our curve coefficients are given by a projective point $(A : C)$ and we can no longer assume that $C = 1$). In practice, it is customary to use 4-isogenies instead where the codomain cost $4S$ and images $2S + 6M$.

In the twisted theta model, a 2-isogeny codomain costs $2S$, an image costs $2S + 2M$, and doubling cost $4S + 4m_0$.

Again, in a Montgomery model with full 2-torsion, it is possible to use an hybrid version, with an image costing $2S + 2m_0$ (translated image from the point of view of the Montgomery model) and doubling costing $2S + 2M + 2m_0$ (translated doubling from the point of view of the twisted theta model).

This lines up the cost of two 2-isogeny with the cost of a 4-isogeny (and is actually slightly better). However, for a 2^n-isogeny chain, it is still better to split into 4-isogenies since this gain on the codomain computations.

2.2. **A time/memory trade off for scalar multiplication on Kummer lines.** We have a time/memory trade off for a Montgomery model with full two-torsion, where we precompute some points to speed up scalar multiplications.

We first start with a precomputation depending only on the base point P and which cost of $2S + 1m_0$ by bits (+ the storage of 2 coefficients by bits). Then a scalar multiplication is in $4M + 2S + 1m_0$ by bit (whether the base point is normalised or not).

The total cost, including the precomputation, is $4M + 4S + 2m_0$ which makes it slightly better than the standard Montgomery ladder (and saves $1M$ on non normalised points).

We can do more precomputions by using 1 global inversion and $2S + 1m_0 + 4M$ by bit, then the following scalar multiplications will cost $3M + 2S + 1m_0$ by bit.

A similar algorithm works in higher dimension. For a Kummer surface the precomputation step costs one global inversion and $12M + 4S + 3m_0$ by bits, and then a scalar multiplication with the same base point costs $7M + 4S + 3m_0$; compared to $7M + 12S + 9m_0$ or $10M + 9S + 6m_0$ for the standard ladder.

2.3. **Pairings on Kummer lines.**

2.3.1. *Generic pairings.* Isogeny based cryptography rely on generic pairings, where we cannot assume that one point lives in a smaller field. In [CLN16], the generic cost of the Tate pairing then becomes $5S + 15M$ for doublings, and $4S + 20M$ for additions (see [Rei23]); much more expansive than a simple scalar multiplication. The best generic algorithm in the litterature, in [BELL10], uses $10M + 9S$ for doubling, and $11.5M + 3S$ by addition.

We work out the arithmetic of the biextension associated to the divisor $2(0_E)$ on the Montgomery model of a Kummer line with full rational 2-torsion. We derive from this an efficient ladder like algorithm for pairings computation. Our ladder algorithm costs $7S + 9M$ by bit, which is closer to the cost of a scalar multiplication via the Montgomery ladder. As special cases, when $n = 2^m$ or we compute a self pairing, the cost goes down to $4S + 6M$ by bit.

We also explain how to compute a standard exponentiation (rather than a ladder) in the biextension, this allows to use window-NAF methods. Our algorithm (for now only in the

theta model) costs $5S + 6M$ for a doubling, and $6S + 24M$ for an addition.[1] This suggests that the second algorithm will be faster than the first one when using a window $w \geq 5$ (or when computing pairings between points of 2^n-torsion).

2.3.2. *Pairing based cryptography.* For pairing based cryptography on elliptic curves, it is convenient to use the Tate pairing with $P \in \mathbb{G}_1 \subset E(\mathbb{F}_q)$, $Q \in \mathbb{G}_2 \subset E(\mathbb{F}_{q^k})$, and k even to allow for denominator elimination.

Counting only operations involving the big field $\mathbb{F}_{q^k}$, Miller's algorithm cost $1M+1S+1m$ by doubling, and $1M+1m$ by addition. Here $1m$ denotes a multiplication between a coefficient in $\mathbb{F}_q$ and a coefficient in $\mathbb{F}_{q^k}$.

When denominator elimination is not possible (because k is odd or Q is not in $\mathbb{G}_2$), the cost becomes $2M + 2S + 1m$ by doubling, and $2M + 1m$ by addition.

When P is in the small field $E(\mathbb{F}_q)$ and Q is in the big field $E(\mathbb{F}_{q^k})$, our Tate pairing algorithm costs (counting only operations in the big field) $2S + 1M + 2m$ by bits. This is competitive with the standard Miller's algorithm, except when denominator elimination is available.

2.4. **Monodromy leak: Projective coordinates leak revisited.** Assume that we are doing a scalar multiplication via the Montgomery ladder: we start with $P = (x_P : 1)$ and compute $Q = n.P = (X_Q : Z_Q)$. In practice, during the ladder we work with affine coordinates (X_Q, Z_Q) rather than projective coordinates (which would imply one division at each step, or at least scaling by a random scalar). It is only at the end of the computation that a division is computed and the coordinate $x_Q = X_Q/Z_Q$ is returned.

A projective coordinates leak happens whenever an attacker can retrieve (X_Q, Z_Q) directly. It was shown in [NSS04] how to use a projective coordinates leak to retrieve a few bits of the secret scalar n. This was revisited in [AGB20] to adapt it to the Montgomery ladder, still recovering only a few bits.

Instead, we can use the formula from the biextension arithmetic (more precisely, we use the cubical torsor structure, a refinement of the biextension arithmetic) to fully recover the secret n via:

- Solving some DLPs in $\mathbb{F}_q^*$
- Solving a degree 2 equation in $\mathbb{Z}/(q-1)\mathbb{Z}$.

In most cases (except if $q-1$ has a lot of prime divisors) this can be done in subexponential time. The name monodromy leak comes from the fact that the biextension arithmetic and cubical torsor structure gives the monodromy information underlying the Tate and Weil pairing.

3. Models

3.1. **The Montgomery model.** The Montgomery model of E is rational whenever there is a rational cyclic subgroup of order 4 in E, i.e. a point R_1 of order four which is rational in the Kummer line (i.e. $\pi(R_1) = \pm R_1$), i.e. there is a point of order 2 T_1 with trivial self Tate pairing. The Montgomery model is the model where R_1 is sent to $(1:1)$, $T_1 = 2R_1$ to $(0:1)$ and 0_E to infinity.

The ramification is given by $(0_E) = (1:0)$, $T_1 = (0:1)$, $T_2 = (A^2 : B^2)$, $T_3 = T_1 + T_2 = (B^2 : A^2)$. Here, T_2, T_3 are not necessarily rational, we denote their coordinates by $(A^2 : B^2)$ to make the link with the theta model more explicit later. Conversely, a Kummer

[1] Standard additions are not available on a Kummer line, but we can compute them over the biextension!

line whose neutral point is at infinity, a ramification point is $(0:1)$, and the remaining two ramifications points are invariant under $(X:Z) \mapsto (Z:X)$ gives a Montgomery model.

Above the canonical two torsion point $T_1 = (0:1)$, we have the canonical four torsion points $R_1 = (1:1) = R_1 + T_1, R_1' = (-1:1) = R_1 + T_2 = R_1 + T_3$.

If T_2 is rational, its coefficients are enough to represent E. The translation by T_2 is given by $(x:z) \mapsto (A^2x - B^2z : B^2x - A^2z)$. From this we can recover the curve coefficient $\mathcal{A}$ of the Montgomery model by $\mathcal{A} = (A:C) = (A^4 + B^4 : -A^2B^2)$, $(\mathcal{A}+2)/4 = ((A^2 - B^2)^2 : -4A^2B^2)$.

In these notes, we will often focus on the arithmetic of the Kummer line E of a Montgomery model with full rational 2-torsion. In this case, the quotient $E' = E/T_1$ is also a Montgomery model with full rational 2-torsion, so we can exploit the symmetry between E and E' in our arithmetic by factorising through the isogeny $f : E \to E'$.

3.2. **Twisted theta models.** The Kummer line associated to a theta model $\theta(a:b)$ has for neutral point $0_E = (a:b)$ and ramification $T_1 = (-a:b), T_2 = (b:a), T_3 = (-b:a)$. We have $R_1 = (1:0), R_1' = (0:1)$ two 4-torsion points above $T_1 = (-a:b)$, and $R_2 = (1:1), R_2' = (1:-1)$ above $T_2 = (b:a)$. We denote $(A^2:B^2) = (a^2+b^2 : a^2-b^2)$.

Conversely, a Kummer line with two rational points of 4-torsion R_1, R_2 such that $T_1 = 2R_1 \neq T_2 = 2R_2$ admits a rational theta model. Equivalently, there are two cyclic subgroups of degree 4 on E, K_1, K_2 such that $K_1 \cap K_2 = \{0_E\}$.

We recall that a Montgomery model can be constructed as long as we have a point of 4-torsion on the Kummer. In theta we have two such points: $(1:0)$ above $(-a:b)$ and $(1:1)$ above $(b:a)$, so we have two associated models. Conversion formula are given in Section 3.4.

From a theta model, we explain how to construct several twisted theta models. When we have a theta model $\theta(a:b)$, we can use the dual theta coordinates given by the Hadamard transform, let me denote that by $\theta'(a':b')$ with $(a':b') = (a+b : a-b)$. We can also twist the theta model by looking at the coordinates $(ax:bz)$ instead of $(x:z)$, let me call this $\theta tw(a^2:b^2)$. We can combine the Hadamard transform and the twisted models to obtain four kind of twisted theta models: $\theta tw, \theta tw', \theta' tw, \theta' tw'$.

A theta model on E arises from a (symmetric) isomorphism of the Heisenberg group of level 2 with the theta group $G(2(0_E))$. In a twisted theta model we take an isomorphism from a twist of the Heisenberg group, so we are on the same elliptic curve, it is the theta structure which is twisted. We will use the model $\theta tw'$, the conversion from θ is $(x:z) \mapsto (ax + bz : ax - bz)$; and the twisted model $\theta' tw'$, the conversion from θ is $(x:z) \mapsto (a'x' + b'z' : a'x' - b'z')$ where $(x':z') = (x+z : x-z)$. We will see that $\theta' tw'$ is (up to translation) the Montgomery model associated to the four torsion point $(1:1)$, and the Montgomery model (up to translation) corresponding to the four torsion point $(1:0)$ is given by $\theta tw'$.

In the $\theta tw'$ model, the neutral point becomes $0_E = (A^2:B^2)$, the 2-torsion $T_1 = (B^2 : A^2), T_2 = (1:0), T_3 = (0:1)$, and the 4-torsion is $R_1 = (1:1), R_1' = (-1:1)$, and $R_2 = (a':b') = (a+b : a-b), R_2' = (b', a')$. In particular, the 4-torsion point $R_2 = (a':b')$ above $T_2 = (1:0)$ allows to recover $(a:b)$.

It is convenient to see $\theta' tw'$ coordinates as follow. Let start with a theta model $\theta(a:b)$ on an elliptic curve E_1; we have an isogeny $f : E_1 \to E_2$ whose kernel is given by the two torsion point $(-a:b)$. We also have a "contragredient" isogeny $\hat{g} : E_1 \to E_0$ whose kernel is given by $(b:a)$.

If E_0 is given by $\theta(a_0:b_0)$, then $g : E_0 \to E_1$ has kernel $(-a_0:b_0)$. From the isogeny formula [Rob23a], we see that a $\theta' tw'$ coordinate $(u:v)$ for $P \in E_1$ can be written as

$(u : v) = (x^2 : z^2)$ where $(x : z)$ is the theta coordinate of $Q \in E_0 \mid g(Q) = P$. In other words: if we use squares of theta coordinates in E_0 to represent points of E_1 we obtain the $\theta' tw'$ model: $\theta^2_{E_0} = \theta' tw'_{E_1}$. We have $(a_0^2 : b_0^2) = (a'^2 + b'^2 : a'^2 - b'^2) = (a^2 + b^2 : 2aa^2 + b^2 : 2ab)$. A similar interpretation holds for the $\theta tw'$ model.

3.3. **Montgomery and theta models.** In the Montgomery model, the neutral point is $0 = (1 : 0)$, the 2-torsion is $T_1 = (0 : 1), T_2 = (A^2 : B^2), T_3 = (B^2 : A^2)$, and the four torsion is $R_1 = (1 : 1), R'_1 = (-1 : 1), R_2 = (a' : b'), R'_2 = (b' : a')$.

In the $\theta tw'$ model, the neutral point becomes $0_E = (A^2 : B^2)$, the 2-torsion $T_1 = (B^2 : A^2), T_2 = (1 : 0), T_3 = (0 : 1)$, and the 4-torsion is $R_1 = (1 : 1), R'_1 = (-1 : 1)$, and $R_2 = (a' : b') = (a + b : a - b), R'_2 = (b', a')$.

The twisted theta model has the same ramification as the Montgomery model, except the neutral point is $(A^2 : B^2)$ which would be a point of 2-torsion T_2 on the Montgomery model, hence why they differ by translation by T_2: $(x : z) \mapsto (A^2x - B^2z : B^2x - A^2z)$. This sends $(-1 : 1)$ to $(1 : 1)$ and conversely (be careful that due to an unfortunate choice of notations, the R_1 on $\theta tw'$ is sent to R'_1 on Montgomery).

We also have a similar conversion to the Montgomery model on the $\theta' tw'$ model. The two torsion on $\theta' tw'$ is given by $0_E = (a_0^2 : b_0^2), T_2 = (b_0^2 : a_0^2), T_1 = (1 : 0)$ and $T_3 = (0 : 1)$. We also have the four torsion point $(1 : 1)$ above $(b_0^2 : a_0^2)$.

The two torsion gives the ramification on the Kummer line. Now notice how we have exactly the same ramification as the Montgomery model $M : y^2 = x(x - \alpha)(x - 1/\alpha)$ with $\alpha = b_0^2/a_0^2$, except that in our case the neutral point is $(a_0^2 : b_0^2)$ while in Montgomery the neutral point is $(0 : 1)$.

This means that the map $\mathrm{Id} : \theta' tw' \to M$ corresponds to the translation by the two torsion point $T_1 = (1 : 0)$ on the $\theta' tw'$ model and by $T_1 = (a_0^2 : b_0^2)$ on the Montgomery model. Via this translation, the four torsion point $(1 : 1)$ above $(b_0^2 : a_0^2)$ indeed become the four torsion point $(1 : 1)$ above $(0 : 1)$ in the Montgomery model as expected.

This gives the following fact: suitably twisting the theta structure, we obtain conversion formula which are free (ie given by the identity) except that a point P in the twisted theta model will correspond to a point $P + T$ in the Montgomery model for some two torsion point T. If we can get an handle on this translation by T, we can combine the best formula for both models.

This was already used in [BRS23] to construct a hybrid Montgomery ladder combining the best of the theta and Montgomery formula. We now describe a similar approach for isogenies. (It is Nicolas Sarkis who found out that we had a free conversion formula between the two models up to a translation by a point of 2-torsion[2], and I realised we could exploit this for isogenies and the scalar multiplication; the implementation was done by Nicolas.)

3.4. **Conversion formula between the theta model and the Montgomery model in dimension 1.** See also [Rob23a, Appendix A].

Let E/k be an elliptic curve, and $(a : b) = (\theta_0(0_E), \theta_1(0_E))$ be its theta null point. We give formula to convert the theta points $(\theta_0(P) : \theta_1(P))$ into the Montgomery coordinates $(x(P) : z(P))$. The formulas follows by looking at the ramification on the Kummer line on both models, and finding the homography that maps the ramification of one model to the other.

[2]We found out afterwards that this was already done in [HR19]

When the theta null point is rational, the elliptic curve E admits both a rational Montgomery model and a rational Legendre model. They are given by

$$y^2 = x(x-\alpha)(x-1/\alpha) = x(x^2 + \mathcal{A}x + 1)$$

and (up to a quadratic twist, which is harmless because we work on the Kummer line anyway) by

$$y^2 = x(x-1)(x-\lambda).$$

These constants are determined as follows: let $(A : B)$ be the dual coordinates of the canonical 2-isogenous curve (we will only need their square). We have

$$A^2 = a^2 + b^2, B^2 = a^2 - b^2, \tag{1}$$

$$\alpha = A^2/B^2 = (a^2+b^2)/(a^2-b^2), \tag{2}$$

$$\lambda = \alpha^2 = A^4/B^4 = (a^2+b^2)^2/(a^2-b^2)^2, \tag{3}$$

$$\mathcal{A} = -(\alpha + 1/\alpha) = -(\alpha^2+1)/\alpha = -(A^4+B^4)/(A^2B^2) = -2(a^4+b^4)/(a^4-b^4), \tag{4}$$

$$(\mathcal{A}+2)/4 = -b^4/(a^4-b^4). \tag{5}$$

Conversely, from $\mathcal{A}$, we can recover $(a : b)$ via

$$\alpha + 1/\alpha = -\mathcal{A}, \tag{6}$$

$$A^2/B^2 = \alpha, \tag{7}$$

$$a^2 = A^2 + B^2, b^2 = A^2 - B^2, (a^2 : b^2) = (\alpha + 1 : \alpha - 1). \tag{8}$$

We note that if $(a : b)$ is a solution, then $(a : \zeta b)$ also with $\zeta \in \mu_4$, these correspond to different theta structures.

With these constants defined, we can now explain how to convert the points. If $P = (x : z)$ in Montgomery coordinates, then

$$(\theta_0(P) : \theta_1(P)) = (a(x-z) : b(x+z)). \tag{9}$$

Conversely, if $P = (\theta_0 : \theta_1)$, then in Montgomery coordinates

$$(x(P) : z(P)) = (a\theta_1 + b\theta_0 : a\theta_1 - b\theta_0). \tag{10}$$

On the theta model $0_E = (a : b)$, we have a canonical basis of the 2-torsion given by $T_1 = (a : -b)$ and $T_2 = (b : a)$. We have a canonical basis of the 4-torsion given by $T_1' = (1 : 0)$ above T_1 and $T_2' = (1 : 1)$ above T_2. The map above sends T_1 to $(0 : 1)$ in the Montgomery model, T_1' to $(1 : 1)$, T_2 to $(A^2 : B^2)$, T_2' to $(a + b : a - b)$.

So conversely, given a Montgomery curve, the canonical point $T' = (1 : 1)$ of 4-torsion above the 2-torsion point $T = (0 : 1)$ and a second point $T'' = (r : s)$ above another point of 2-torsion, then the theta null point $(a : b)$ induced by the basis (T', T'') of the 4-torsion is given by $(r + s : r - s)$.

For the case of a general elliptic curve E with a basis (T', T'') of the 4-torsion, we first convert E to a Montgomery model by sending T' to $(1 : 1)$ and $T = 2T'$ to $(0 : 1)$, the map is then $x \mapsto (x - x(T))/(x(T') - x(T))$. Then we apply the above formula to the image of T''.

The alternative Montgomery model. When we have a theta model, we can also introduce the dual theta coordinates

$$(\theta'_0 : \theta'_1) = (\theta_0 + \theta_1 : \theta_0 - \theta_1),$$

in particular the dual theta null point is given by $(a' : b') = (a + b : a - b)$. We can construct another Montgomery model by replacing in the above formula $(a, b, \theta_0, \theta_1)$ by $(a', b', \theta'_0, \theta'_1)$.

Plugging in this different model the equations expressing $(a', b', \theta'_0, \theta'_1)$ in terms of $(a, b, \theta_0, \theta_1)$, we obtain alternative formulas:

$$A'^2 = a'^2 + b'^2 = 2(a^2 + b^2), B'^2 = a'^2 - b'^2 = 4ab, \tag{11}$$

$$\alpha' = A'^2/B'^2 = (a^2 + b^2)/(2ab), \lambda' = \alpha'^2, \tag{12}$$

$$\mathcal{A}' = -(\alpha' + 1/\alpha') = -(a^4 + 6a^2b^2 + b^4)/(2(a^3b + ab^3)), \tag{13}$$

$$P = (x : z) \mapsto (\theta_0(P), \theta_1(P)) = (ax - bz : bx - az), \tag{14}$$

$$(\theta_0, \theta_1) \mapsto (x(P) : z(P)) = (a\theta_0 - b\theta_1 : b\theta_0 - a\theta_1). \tag{15}$$

4. Scalar multiplication on Kummer lines

4.1. **Standard arithmetic in the Montgomery and theta models.** Let us first recall the standard formulas in the theta and Montgomery models.

Differential additions in theta coordinates are computed as follow, using [Rob23a, § 5, § 6]. Let $(a : b)$ be the neutral point, and $(A : B)$ as usual: $(A^2 : B^2) = (a^2 + b^2 : a^2 - b^2)$. Let $(XX_P : ZZ_P) = (X_P^2 + Z_P^2 : X_P^2 - Z_P^2)$, $(XX_Q : ZZ_Q) = (X_Q^2 + Z_Q^2 : X_Q^2 - Z_Q^2)$, $(U : V) = (B^2XX_PXX_Q : A^2ZZ_PZZ_Q)$, $(X(P+Q) : Z(P+Q)) = (Z(P-Q)(U+V) : X(P-Q)(U-V))$. Applying this to $P = Q$ gives the doubling.

It is easy to extend these formula to the different twisted variant of the theta model.

In the Montgomery model, the usual differential addition is given as follow: $(U_1 : U_2) = (X_P + Z_P : X_P - Z_P)$, $(U_3 : U_4) = (X_Q + Z_Q : X_Q - Z_Q)$, $(X(P + Q) : Z(P + Q)) = (Z(P - Q)(U_1U_4 + U_2U_3)^2 : X(P - Q)(U_1U_4 - U_2U_3)^2)$.

We cannot apply this for doubling however, because the neutral point in Montgomery is $(1 : 0)$ so we get a division by zero. For doubling we instead use: $(U_1 : U_2) = (X_P + Z_P : X_P - Z_P)$, $U_3 = (U_1^2 - U_2^2)$, $(X(2P) : Z(2P)) = (U_1^2U_2^2 : U_3(U_2^2 + (\mathcal{A} + 2)/4U_3)$. If $\mathcal{A} = (\mathsf{A} : \mathsf{C})$, the last line becomes $(X(2P) : Z(2P)) = (U_1^2U_2^2k_2 : U_3(U_2^2k_2 + k_1U_3)$ with $(k_1 : k_2) = (\mathsf{A} + 2\mathsf{C} : 4\mathsf{C})$.

4.2. **Hybrid arithmetic.** From Section 3.3, if we work in $\theta tw'$ but use the doubling formula in Montgomery to compute $P \to 2P$ instead of the ones of the twisted theta model, we actually obtain $2(P + T_2) + T_2 = 2P + T_2 = 2P + (1 : 0)$ in the $\theta tw'$ model. We can thus compute a translated doubling $2P + T_2$ in $4S + 4m_0$ ($4S + 2m_0$ if we normalize the curve constants), which is interesting if $S < M$ and m_0 is small. We are off by a translation by the point of 2-torsion T_2, but this is easily adjusted to when doing a scalar multiplication by the Montgomery ladder: this does not affect doublings, and for differential additions we just need to track if the base point is P or $P + T_2$.

More generally, given $P, Q, P - Q + T_2$, we can compute $P + Q + T_2$ as follow: $t = (x_P + z_P)(x_Q + z_Q)/(A^2 + B^2)$, $u = (x_P - z_P)(x_Q - z_Q)/(A^2 - B^2)$, $(P + Q + T_2) = ((t + u)^2/x(P - Q + T_2) : (t - u)^2/z(P - Q + T_2))$. This costs $2S + 4M + 2m_0$, $-1m_0$ if the constants are normalised, $-1M$ if the base point $P - Q + T_2$ is normalised.

Applying the formula to $P = Q$, we get the (translated) doubling formula: $t = (x_P + z_P)^2/(A^2 + B^2), u = (x_P - z_P)^2/(A^2 - B^2), (2P + T_2) = ((t + u)^2/A^2 : (t - u)^2/B^2)$, which costs $4S + 4m_0$ ($-2m_0$ if the constants are normalised).

Using these formula, we get the hybrid ladder whose cost is $3M + 6S + 2m_0$.

Likewise, if working in $\theta' tw'$ we use the doubling formula in M to compute $P \to 2P$ instead of the ones of the twisted theta model, we actually obtain $2(P+T_1)+T_1 = 2P+T_1 = 2P + (1 : 0)$ in the $\theta tw'$ model. The doubling formula on M requires $\mathcal{A} = -\alpha - 1/\alpha$ so that $M : y^2 = x(x^2+\mathcal{A}x+1)$; more precisely it requires $(\mathcal{A}+2 : 4) = ((a_0^2-b_0^2)^2 : -(a_0^2+b_0^2)^2)$ so can be computed in $2S + 2a$ from $(a_0^2 : b_0^2)$.

5. 2-isogenies between Kummer lines

5.1. **Standard isogeny formulas.** In the Montgomery model, for an isogeny with kernel $T_2 = (A^2 : B^2) \neq T_1 = (0 : 1)$, the formula is given by [Ren18] $(\mathcal{A}' : 1) = (2(B^4-2A^4) : B^4)$, $(\mathcal{A}' + 2 : 4) = (B^4 - A^4 : B^4)$, and images are given by $(X : Z) \mapsto (X(XA^2 - ZB^2) : Z(XB^2 - ZA^2))$.

In the theta model, the isogeny with kernel $T_1 = (a : -b)$ can be written as follow [Rob23a, § 15.1]. Let $T = (r : s) \in E_0$ be a 8-torsion point above the 4-torsion point $R_1 = (1 : 0)$ which itself is above the 2-torsion point $T_1 = (-a_0 : b_0)$. Then $(A : B) = (r^2 + s^2 : r^2 - s^2)$ so $(a_2 : b_2) = (r^2 : s^2)$. And the isogeny $\theta_{E_1} \to \theta'_{E_2}$ is given by $(x : z) \mapsto (B(x^2 + z^2) : A(x^2 - z^2))$; we need an Hadamard transform to obtain the coordinates in θ_{E_1}.

From these isogeny formula, we can recover isogeny formula on our twisted models and also on the Montgomery model from applying base change. We explain a more general method in Section 5.2.

Let us show how to use the second method, i.e. by base change. To express the isogeny $f : E_1 \to E_2$ in the models $\theta' tw'_{E_1} \to \theta' tw'_{E_2}$ corresponds to writing the isogeny in the models $\theta^2_{E_0} \to \theta^2_{E_1}$. By the above description, the isogeny $g : E_0 \to E_1$ can be written as follow. Let $T = (r : s) \in E_0$ be a 8-torsion point above the 4-torsion point $(1 : 0)$ which itself is above the 2-torsion point $(-a_0 : b_0)$. Then $(a' : b') = (r^2 + s^2 : r^2 - s^2)$ so $(a : b) = (r^2 : s^2)$. And the isogeny $\theta_{E_0} \to \theta'_{E_1}$ is given by $(x : z) \mapsto (b'(x^2 + z^2) : a'(x^2 - z^2))$ so we need an Hadamard transform to obtain the coordinates in θ_{E_1}.

We can use this to describe the isogeny $\theta^2_{E_0} \to \theta^2_{E_1}$. The neutral point on E_1 described by the $\theta^2_{E_0} = \theta' tw'_{E_1}$ is $(a_0^2 : b_0^2)$. The point T above corresponds to a 4-torsion point $T = (r^2 : s^2)$ on $\theta^2_{E_0}$ above the two torsion point $(1 : 0)$ (we can check that in $\theta' tw'_{E_1}$ $(1 : 0)$ corresponds to the 2-torsion point $(-a : b)$ in θ_{E_1}).

Then we can compute $(a : b) = (r^2 : s^2)$, $(a' : b') = (a + b : a - b)$ and the neutral point of E_2 in the $\theta' tw'_{E_2} = \theta^2_{E_1}$ model is $(a^2 : b^2)$ can be computed in $2S$, $2S + 2a$ if we include $(a' : b')$ which will be needed for images.

Let $P = (x^2 : z^2) \in E_1$ in the $\theta^2_{E_0}$ model. The image of P in E_2 in the $\theta^2_{E_1}$ model can be computed as follows: compute $(b'(x^2 + z^2) : a'(x^2 - z^2))$ apply the Hadamard transform and then square the coordinates; this costs $2S + 2M + 4a$.

This gives the isogeny algorithm in the θ^2 model, it is also well known how to compute doublings in this model, see [BRS23] for more details.

A similar approach gives formula in the $\theta tw'$ model. In the $\theta tw'$ model, the isogeny f with kernel by T_1 is given by $(x : z) \mapsto ((x + z)^2/a^2 : (x - z)^2/b^2)$. The neutral point of

E' is then $(a^2 : b^2)$, T_2, T_3 are mapped to $(b^2 : a^2)$, R_1 is mapped to $(1 : 0)$, R_1' to $(0 : 1)$, R_2, R_2' to $(1 : 1)$. The dual isogeny $\tilde{f}$ is given by $(x : z) \mapsto ((x + z)^2/A^2 : (x - z)^2/B^2)$.

And the isogeny with kernel T_2 is given by $g : (x : z) \mapsto (((x + z)/a + (x - z)/b)^2 : ((x+z)/a-(x-z)/b)^2)$. We recall that $(a : b)$ can be recovered from R_2. The neutral point is then $g(0) = g(T_2) = (a'^2 : b'^2), g(T_1) = g(T_3) = (b'^2 : a'^2), g(R_1) = g(R_1') = (1 : 1)$, $g(R_2) = (1 : 0), g(R_2') = (0 : 1)$.

5.2. **A general framework to derive 2-isogenies between Kummer lines.** Now we want to extend these classical formulas to more general models.

In dimension one, we can work on any model of a Kummer line by specifying its ramification (+ the neutral point). From this data it is easy to recover the action of the theta group $G(2 0_E)$, and hence compute formula for 2-isogenies between two models. This also allows to obtain doubling formula, and by considering the isogeny: $(P_1, P_2) \mapsto (P_1 + P_2, P_1 - P_2)$ differential addition formula.

If T is a rational two torsion point on our model, we can consider the action g_T of a rational element $g_T \in G(2 0_E)$ in the theta group above T on the sections $\Gamma(E, 2 0_E) = \langle X, Z\rangle$. This action is irreducible and faithful, so g_T is completely determined by this action. Then $\lambda = g_T^2 \in \mathbb{G}_m$ is an element, and its class in $k^*/k^{*,2}$ does not depend on the representative, only on T. A small computation shows that this is exactly the (non reduced) Tate pairing $e_{T,2}(T, T)$. The symmetric elements above T are of order exactly 2, so $e_{T,2}(T, T)$ is trivial precisely when these symmetric elements are rational.

We remark that the translation by T is given by a projective homography, which can be determined by the fact that it maps $0 \mapsto T, T \mapsto 0, T_2 \mapsto T_3, T_3 \mapsto T_2$; and we can take for g_T any rational affine lift of this projective translation.

By an homography, we can always send 0_E to $(1 : 0)$ and T to $(0 : 1)$. An element g_T can be given in the form $(X : Z) \mapsto (Z : \lambda X)$, so if $T_2 = (x_2 : z_2)$, $T_3 = (x_3 : z_3)$, so $\lambda = x_3 z_3/x_2 z_2$, this is well defined in $k^*/k^{*,2}$. Notice that T_2, T_3 are projectively determined only up to an homotety, but this does not change the class of λ.

We can also describe the two points of 4-torsion above T (rember that we are on the Kummer, so $[T' + T] = [T']$ by solving the equation $T' + T = T'$.

From all this discussion, it follows that $\lambda = 1$ iff the symmetric elements $\pm g_T$ are rational iff $e_{T,2}(T, T) = 1$, iff the curve is of Montgomery type when T is sent to $(0 : 1)$ and 0_E to infinity, iff (still if these two points are sent likewise) $T' = (1 : 1)$ is a point of 4-torsion above T, iff there are sections X, Z of $2(0_E)$ such that $g_T(X, Z) = (Z, X)$ iff (by Hadamard) there are sections such that $g_T(X, Z) = (-X, Z)$, iff the quotient E/T is of Legendre type. In particular, if T is of Montgomery type and the full 2-torsion is rational, then E/T has full 2-torsion and the generator of the dual isogeny is of Montgomery type, which is a nice symmetric situation.

Anyway, to study the 2-isogeny with kernel $T, f : E \to E' = E/T$, we need to descend the divisor $4(0_E)$ to $2(0_{E'})$. Since the descent is symmetric, it is given by one of the symmetric element $H_T \in G(4(0_E))$ above T, and it is not hard to prove that it is the symmetric element which is given by $H_T = h_T^{\otimes 2}$ where $\pm h_T$ is any of the two symmetric element above T in $G(2(0_E))$. Although h_T may not be rational, if we have g_T of type λ, then by definition of λ, $H_T = g_T^{\otimes 2}/\lambda$, so H_T is always rational.

It follows that the elements of $\Gamma(2(0_{E'}))$ are the sections of $\Gamma(4(0_E))$ invariant by H_T. Now since we only have $(X, Z) \in \Gamma(2(0_E)) = \Gamma(2(0_E))^+$, we can only construct the even elements $(X^2, XZ, Z^2) \in \Gamma(4(0_E)^+)$ (and these span the space of even elements, the rest of $\Gamma(4(0_E))$ is obtained by adding the odd element YZ). Now take X, Z such that $g_T.(X, Z) =$

$(Z, \lambda X)$ as above, then $H_T = g_T^{\otimes 2}/\lambda$ acts on this basis by $H_T.X^2 = Z^2/\lambda, H_T.Z^2 = \lambda X^2, H_T.XZ = \lambda XZ$. Take a space (U, V) of invariants under this action, and compute the ramification of E' by computing (U, V) on T_2, T_3 and T_1'. This give a model of E', then eventually compose by an homography to make it of nice form. All this gives a very general framework to compute formulas for 2-isogenies between different models of Kummer line.

The nicest case is when T is of Montgomery type, ie $\lambda = 1$. If we put T in position $(0 : 1)$ and 0_E to $(1 : 0)$, like in the Montgomery model, then g_T is given by $(X, Z) \mapsto (Z, X)$. We apply the Hadamard change of variable: $(X', Z') = (X + Z, X - Z)$. Via this change of variable, the action of the symmetric g_T is given by $(X', Z') \mapsto (X', -Z')$, so we can take $U = X'^2, V = Z'^2$. This explain why on a Montgomery point, the 2-isogeny images is given by two squares (followed by a nice homography to make E' still of Montgomery type).

There are two ways to ensure that E' is still of Montgomery type. The first one is to ask for T_2, T_3 to be rational; this was handled above. The second one is to ask for a rational point of 8-torsion T''_1 above $T_1' = (1 : 1)$, then $f(T''_1)$ gives a 4-torsion point above $f(T_1')$, hence $f(T_1')$ is still of Montgomery type. Following the above strategy, we get the following isogeny formula: $(X : Z) \mapsto (\gamma(X - Z)^2 : 4XZ)$, with $\gamma = (4rs : (r - s)^2)$ where $T''_1 = (r : s)$. (Recall that $4XZ = (X + Z)^2 - (X - Z)^2$). We have $f(0) = f(T_1) = (1 : 0), f(T_2) = f(T_3) = (1 : -\gamma), f(R_1) = (0 : 1), f(R_2) = (-\gamma : 1)$. We recover formulas from [DJP14].

The reader can check that we can recover all formulas from Section 5.1 this way. From these, we can recover the standard doubling addition and differential addition via Montgomery's formula, they cost $2M+2S+2m_0$ and $4M+2S$ respectively. These cost drop to $2M+2S+1m_0$ if we normalize the constants to have $C = 1$ and to $3M + 2S$ if the base point $P = (X_P : Z_P)$ is normalised to have $Z_P = 1$.

Example 5.1. Let's explain how to compute an isogeny from a theta model to a Montgomery model when we do not have access to a 8-torsion point. From the theta null point $(a : b)$ of E_1, we can compute $(a^2 : b^2)$ the theta null point of E_2 in the $\theta^2_{E_1} = \theta' t w'_{E_2}$ model, and the isogeny map is $(x : z) \mapsto (x^2 : z^2)$. Translating by $T_2 = (1 : 0)$ we obtain the coordinates on the Montgomery model of E_2, with $A_2 = -\alpha_2 - 1/\alpha_2, \alpha_2 = b^2/a^2$.

5.3. **Translated isogenies.** There are two ways to obtain translated isogeny formulas. The first one is to remark that computing isogenies in the $\theta t w'$ model (or $\theta' t w'$ model) is slightly faster than in Montgomery, but doubling is faster in Montgomery. Then we can apply the same strategy as in Section 4.2 and work in (say) the Montgomery model but apply the isogeny formula from $\theta t w'$, which from the point of view of the Montgomery model looks like a translated isogeny.

The second way is to apply the method of Section 5.2, find invariant sections, look at the image ramification, but not translate back to send the isogeneous neutral point to infinity. Since we skip the translation, we get faster formula, but if we work in the new codomain as if the point at infinity was our neutral point, we are off by translation by some point. This method is more generic (it applies to all models), but of course give back the same formula as the first model. We will illustrate both. As an aside, translated isogenies allows to recover translated doublings too (by applying the translated dual isogeny). This gives an alternative way to recover the formula from Section 4.2 directly on a Montgomery model, without going through the change of variable to the twisted theta model.

Let's first look at what happens in the Montgomery model where we use the theta formula for images, using the model $\theta' t w'$. As explained above, the point $(a_0^2 : b_0^2)$ corresponding to the neutral point in the $\theta' t w'$ model now corresponds to a 2-torsion point T_1 in M. We can represent M by this 2-torsion point, for doubling we need $(A + 2 : 4) = ((a_0^2 - b_0^2)^2 :$

$-(a_0^2+b_0^2)^2)$ which we can recover in $2S$ from T_1. The equation is given by $M : y^2 = x(x-\alpha)(x-1/\alpha) = x(x^2+\mathcal{A}x+1)$ with $\alpha = b_0^2/a_0^2$.

We want to compute the isogeny $E_1 \to E_2$ with kernel T_1. Here E_1 is in a Montgomery model where the full 2-torsion is rational, and we quotient by a 2-torsion point which is different from $(0:1)$, so that the four torsion point $(1:1)$ in E_1 is still of four torsion in E_2 and E_2 still has a Montgomery model.

But we want to represent E_2 via a two torsion point like we did for E_1 (more precisely the two torsion point giving the next kernel). So we need to assume that we have a 4-torsion point $T = (r^2 : s^2)$ above T_1. Then $f(T) = T_1' := (r^4 : s^4)$ on E_2 is the two torsion point we use to represent E_2 and is computed in $2S$. This point T_1' will be the kernel of our next isogeny.

From the Montgomery point of view, given a point $P = (x^2 : z^2)$, then computing $(b'(x^2+z^2) : a'(x^2-z^2))$ followed by the Hadamard transform then squaring the coordinates corresponds to computing $f(P)+T_1'$ on E_2 and can be done in $2S+2M+4a$. Since T_1' is the kernel of the next isogeny, it does not matter that we translate the image, except at the very last step.

In other words: if on a Montgomery curve we have a point of four torsion T which does lies above a two torsion point $T_1 \neq (0:1)$, then if f is the isogeny with kernel T_1 we can use the coordinates of T to compute $P \mapsto f(P)+f(T)$ in $2M+2S+4a$ compared to $4M+4a$ for computing $f(P)$. And $f(T)$ can be computed in $2S$.

For a 2^n-isogeny, if we select T to be the point giving the next kernel, this extra translation in images does not matter (except at the last step).

Now, we reexplain how to get formula by working on the Montgomery model directly, but using the methods of Section 5.2. Let's look at the isogeny $f : E \to E/T_1$, we have seen that invariant sections are given by $U = (X+Z)^2, V = (X-Z)^2$. Let's look at the images of the ramification and 4-torsion points under $G = (U, V)$. First we apply the Hadamard transform: $H(0_E = (1:0)) = (1:1), H(T_1 = (0:1)) = (-1:1), H(T_2 = (A^2 : B^2)) = (a^2 : b^2), H(T_3 = (B^2 : A^2)) = (-a^2 : b^2), H(R_1 = (1:1)) = (1:0), H(R_1' = (-1:1)) = (0:1), H(R_2 = (a' : b')) = (a:b), H(R_2' = (b' : a')) = (-a:b)$.

It follows that $G(0_E) = g(T_1) = (1:1), G(T_2) = g(T_3) = (a^4 : b^4), G(R_1) = (1:0), G(R_1') = g(R_1+T_2) = (0:1), G(R_2) = g(R_2') = (a^2 : b^2)$. The ramification on the codomain is $(1:1), (a^4 : b^4), (1:0), (0:1)$, given as the image of $0_E, T_2, R_1, R_1'$. We can scale it to be invariant by $(X:Z) \mapsto (Z:X)$ as in the Montgomery form, the scaling is $(X:Z) \mapsto (b^2X : a^2Z)$ and the ramification is then $(b^2 : a^2), (a^2 : b^2), (1:0), (0:1)$. However, the point 0_E is sent to $T_2' := (b^2 : a^2)$. So in summary, if H is the Hadamard transform and S the squaring $(X:Z) \mapsto (X^2 : Z^2)$ transform, and C the scaling transform above, we have that $C \circ S \circ H$ is an isogeny with kernel T_1 between our Montgomery curve, and a curve M'' that has the same ramification as a Montgomery curve M' except the neutral point is $T_2' = (b^2 : a^2)$. So the full isogeny, if we want to work on M' rather than M'', is given by translating by T_2'; in other word $C \circ S \circ H : M \to M'$ gives the isogeny translated by $T_2' = f(R_1)$.

In summary, applying the above method, we get the following formulas.

The (translated) isogeny $f : E \to E/T_1$ is given by, if $P = (x:z)$, $f(P+R_1) = ((x+z)^2/a^2 : (x-z)^2/b^2)$, with $(a^2 : b^2) = (A^2+B^2 : A^2-B^2)$. Notice that $E' = E/T_1$ is still a Montgomery curve with full rational 2-torsion, so there is a perfect symmetry between E and E'. We have $T_2' = f(R_1) = (b^2 : a^2), f(0) = f(T_1) = (1:0) = 0$, $f(T_2) = f(T_3) = (0:1) = T_1', f(R_1) = T_2' = (b^2 : a^2), f(R_2) = (a^2 : b^2) = T_3'$.

The (translated) dual isogeny with kernel T_1' is given by $\tilde{f}(P + R_1') = \tilde{f}(P) + T_2 = ((x+z)^2/A^2 : (x-z)^2/B^2)$. Composing $\tilde{f} \circ f$ we recover the (translated) doubling formula $P \mapsto 2P + T_2$ as above. We have $\tilde{f}(0) = \tilde{f}(T_1') = (1:0), \tilde{f}(T_2') = \tilde{f}(T_3') = T_1, \tilde{f}(R_1') = T_2 = (A^2 : B^2), \tilde{f}(R_2') = (B^2 : A^2) = T_3$.

Now, let $g : E \to E_2 = E/T_2$ be the isogeny with kernel T_2. Since we want E_2 to be Montgomery with full rational two torsion, we need a point $S_2 = (a' : b')$ above T_2.

The isogeny g is then given by $g(P + S_2) = (((x+z)/a + (x-z)/b)^2 : ((x+z)/a - (x-z)/b)^2)$, with $(a:b) = (a' + b' : a' - b')$. (Remark that $(A^2 : B^2) = (a^2 + b^2 : a^2 - b^2) = (a'^2 + b'^2 : 2a'b')$). The curve E_2 is represented by its two torsion point $T_2' = g(S_2) = (a'^2 : b'^2)$. The codomain computation costs $2S$, and a translated image $2S + 2m_0$.

We have $g(0) = g(T_2) = 0, g(T_1) = g(T_3) = T_1' = (0:1), g(R_1) = g(R_2) = R_2' = (1:-1), g(R_1 + S_2) = R_1' = (1:1), g(S_2) = T_2' = (a'^2 : b'^2), g(S_2 + T_1) = g(S_2 + T_3) = (b'^2 : a'^2) = T_3'$. The dual isogeny $\tilde{g}$ has kernel $T_1' = (0:1)$ and is given by $\tilde{g}(P) = (B^2(x+z)^2 : 4A^2xz)$. (Notice that $4xz = (x+z)^2 - (x-z)^2$ so $\tilde{g}$ can be computed in $2S + 2m_0$.) We have $\tilde{g}(0) = \tilde{g}(T_1') = (1:0), \tilde{g}(T_2') = \tilde{g}(T_3') = T_2, \tilde{g}(R_1') = T_3, \tilde{g}(R_2') = T_1$. The composition $\tilde{g} \circ g$ gives an alternative formula to compute $P \mapsto 2P + T_2$ in $4S + 2m_0$.

5.4. **Theta versus Montgomery.** To summarize, the complexities for computing isogenies in the theta model are as follows:

(1) $2S + 2a$ for the codomain
(2) $2S + 2M + 4a$ for an image
(3) $4S + 4M + 8a$ for doubling

The input is the theta null point $(a:b)$, which implicitly contains the 2-torsion point $(-a:b)$ used for our kernel; and the images computations needs (some constants computed during) the codomain.

In the Montgomery model, the costs are, using [CLN16; CH17; Ren18]:

(1) $2S + 1a$ for the codomain
(2) $4M + 4a$ for an image (using a precomputation of $2a$)
(3) $2S + 4M + 4a$ for doubling

Here the input is a two torsion point (different from $(0:1)$) giving the kernel (and implicitly the curve); the image computation does not needs the codomain.

We see that the theta model is slightly faster then the Montgomery model except for doublings. Using hybrid isogenies allows to combine the best of both models.

To sum up, we can, provided we have a point of 4-torsion T above our kernel $\langle T_2 \rangle$:

(1) Compute a representation of the codomain in $2S$. The representation is given by the 2-torsion point $f(T) = T_2$, which is the kernel of the next isogeny.
    If we need to compute doublings on the codomain, we need to add a $2S + 2a$ precomputation to compute $(\mathbb{A} + 2 : 4)$, and if we need to compute images we need to add a $2a$ precomputation (which is already done if we did the previous $2S + 2a$ precomputation needed for doublings).
(2) Compute "images" in $2M + 2S + 4a$.
(3) Compute "doublings" in $4M + 2S + 4a$.

The words "images" and "doublings" are in quotes because if we consider that we are on a twisted theta models the "doublings" we compute are actually $2P + T_2$, while if we consider that we are in the Montgomery model it is the images that are actually given by $f(P) + T_2$.

The images need some of the constants computed for the codomain. In both cases, this translation is by an element of the next kernel, so does not affect the rest of the computation.

We conclude this with a discussion on 4-isogenies. On the Montgomery model, a 4-isogeny can be computed in [CH17]:

(1) $4S + 5a$ for the codomain
(2) $6M + 2S + 6a$ for images.

Here the input for the codomain is given by the coordinates of a 4 torsion point T, and the input for the images needs some of the constants for the codomain.

If the kernel is given by $K = \langle T \rangle$, we can also look at the cost of decomposing this 4-isogeny as a 2-isogeny where we exploit the 4-torsion point followed by a standard 2-isogeny formula in the Montgomery model:

(1) $4S + 3a$ for the codomain
(2) $6M + 2S + 8a$ for images.

It is probable (but I haven't checked) that we actually obtain essentially the same formula as the 4-isogeny algorithm above, except 2 of the additions needed for images could be moved to a precomputation done in the codomain computation. So the standard 4-isogeny formula can essentially be interpreted as alternating the $2M + 2S$ isogeny formula with the $4M$ isogeny formula. But if we have a 2^n-isogeny to compute, we might as well keep using the $2M + 2S$ formula, except at the very end where we use the $4M$ formula to not be off by translation by a point of 2-torsion (and we might not have a 4-torsion point available anymore anyway).

Remark 5.2. Decomposing a 2^n-isogeny via 2-isogenies or 4-isogenies. While the above formula for 2-isogenies in the Montgomery model are fun to look at, they are not really useful in practice: it is better to decompose a 2^n-isogenies as a sequence of 4-isogenies rather than as a sequence of 2-isogenies. The reason is that the decomposition algorithm is quasi-linear, if we split into blocks of 2^m-isogenies, we gain a bit more than m images and doublings because of the quasi-linearity. Usually this is not interesting because a 2^m-isogeny costs $O(2^m)$ to compute, so is 2^{m-1} more expansive than a 2-isogeny but $2^2 = 4$ hits the sweet spot for an optimal decomposition time. Once an isogeny is decomposed, for a 2^n-image, using the slightly faster 2-isogeny images rather than 4-isogenies would be better however.

6. Time-Memory trade off for the arithmetic

I found these formula (first for the theta model) in December 2022, while working with Barbulescu and Sarkis on models of Kummer lines.

6.1. **Overview.**

Although the Montgomery ladder is very efficient, for fast scalar multiplication the twisted Edward model is often faster because it allows for a time/memory trade off by using the window-NAF method to reduce the amount of additions.

However, when the scalar is a secret, these time/memory trade off are often susceptible to side channel attacks, so although signing on Curve25519 is implemented in Edwards coordinate, the DH key exchange uses the Montgomery ladder.

It might seem that a time/memory trade off is not possible on a Kummer line because standard additions are not available. A way to precompute the Montgomery ladder was presented in [OLHFR18]. When T_2 is rational, we present a novel approach to precompute the ladder that:

(1) does a precomputation of points $P_i = (X_i : Z_i)$ costing $2S + 1m_0$ by bit, and requiring to store two field coefficients by bit.

(2) using this precomputation, the ladder then costs $2S + 1m_0$ for doubling, and $4M$ for a differential addition by bit.

The total cost, including the precomputation, is thus of $4S + 4M + 2m_0$, and further scalar multiples with the same base point then cost $2S + 4M + 1m_0$. Here it does not matter whether $P = (X_P : Z_P)$ is normalised or not.

We stress that the scalar multiplication still uses a ladder approach, with one doubling and one differential addition by bit, thus retaining the same side channel resistance as the standard Montgomery ladder. (We recall that the Montgomery ladder without precomputation costs $5M + 4S + 1m_0$ when the base point P is normalised, and $6M + 4S + 1m_0$ if P is not normalised.)

If we know that P will be used several time (like for the first step of a DH key exchange), we can increase the precomputation to normalise the points $P_i = (X_i/Z_i : 1)$. This costs one field division by bit, and reduces the storage to one field coefficient by bit. We can batch the inversions, to replace the one division by bits by one global inversions and $3 + 1 = 4$ multiplications by bit.

The precomputation is then one global inversion, and $4M + 2S + 1m_0$ by bit (the storage drops to one coefficient by bit). The multiples $m.P$ then cost $2S+3M+1m_0$ by bit, significantly improving on the standard ladder.

Unfortunately, for Curve25519 the point T_2 is not rational. But its 2-isogeneous curve is a Montgomery curve with full rational 2-torsion, so by computing an isogeny at the beginning and the end we can still use our novel time/memory trade off on Curve25519 (however, unlike Curve25519, the curve constant on the isogeneous curve is not small, so we don't gain as much as if we had selected from the beginning a suitable curve with a small m_0).

A similar algorithm works in higher dimension. For a Kummer surface the precomputation step costs one global inversion and $12M + 4S + 3m_0$ by bits, and then a scalar multiplication with the same base point costs $7M + 4S + 3m_0$; compared to $7M + 12S + 9m_0$ or $10M + 9S + 6m_0$ for the standard ladder.

Now let us compare with the results of [OLHFR18]. Their idea is to use a right to left Montgomery ladder rather than the usual left to right ladder. The right to left ladder always involve the points $P_i = 2^iP$ (where P is the base point), so these can be precomputed. (Our approach is related: we remark that we can factor doublings and differential additions through 2-isogenies to get half doublings and half differential additions. We permute the order: rather than at each step, compute an image through a 2-isogeny and then do a half differential additions, we precompute all images and then only do half differential additions for the ladder).

With these precomputations done, the right to left Montgomery ladder then only needs differential additions, except that the difference is not fixed anymore. So a priori we need a full DiffAdd at each step rather than a mDiffAdd. In [OLHFR18], the authors explain how to extract from the P_i a coordinate μ_i (this requires a division) which can be used to get a DiffAdd formula (involving P_i) in $3M + 2S$, exactly like the mDiffAdd.

Their precomputation cost is then the cost of one doubling by bit (to compute the 2^iP, aka $2M + 2S + 1m_0$ and one division, so batching inversions we get a precomputation cost of one global division and $6M + 2S + 1m_0$ by bit. The scalar multiplication is then like the standard ladder, except all doublings have been removed and only differential addition remains, so it costs $3M + 2S$ by bit. Without the computation of the μ_i, the precomputation would be $2M + 2S + 1m_0$ and the multiplication cost $4M + 2S$ by bit.

We compare two cases. We recall that the standard Montgomery ladder costs $5M + 4S + 1m_0$ by bit when P is normalised.

- We only do a light precomputation without inversions. By bit, our ladder requires a $2S + 1m_0$ precomputation, followed by a $4M + 2S + 1m_0$ for multiplication. The ladder in [OLHFR18] requires a $2M + 2S + 1m_0$ precomputation, followed by $4M + 2S$ for multiplication.
- We allow a global inversion in the precomputations. By bit, our ladder requires a $4M + 2S + 1m_0$ precomputation, followed by a $3M + 2S + 1m_0$ for multiplication. The ladder in [OLHFR18] requires a $2M + 2S + 1m_0$ precomputation, followed by $3M + 2S$ for multiplication.

We see that in both cases, our precomputation is smaller, but we pay for it by needing an extra m_0 in the multiplication step. However, an important point is that our light precomputation is so cheap that even including it we are only at $4M + 4S + 2m_0$, which gains $1M - 1m_0$ compared to the Montgomery formula (and $2M - 1m_0$ if the base point is not normalised).

Furthermore, for Kummer surfaces, our precomputation and multiplication are both faster than an equivalent approach as [OLHFR18] would provide.

6.2. **Explicit formula.** In the theta model, the arithmetic ladder stems from the duplication formula (see [Rob23a]): $\theta_E(P+Q) \star \theta_E(P-Q) = H(\theta'_{E'}(f(P)) \star \theta'_{E'}(f(Q)))$.

The ladder use two steps for the differential addition (doubling is a special case where $P - Q = 0$): compute $f(P)$ via $\theta_E(P) \star \theta_E(P) = H(\theta'_{E'}(f(P)) \star \theta'_{E'}(f(0)))$. This costs $2S + 1m_0$. Do the same for $f(Q)$. Then use $\theta_E(P+Q) \star \theta_E(P-Q) = H(\theta'_{E'}(f(P)) \star \theta'_{E'}(f(Q)))$ to compute $(P+Q) \star (P-Q)$ in $2M$, and then $P+Q$ in again $2M$ (or $1M$ if $P-Q$ is normalised).

A large part of the ladder is hence spent in isogeny images. Let $f_1 = f, f_2 = \tilde{f} \circ f_1, f_3 = f \circ f_2$, $f_4 = \tilde{f} \circ f_3$ and so on. Assume we had $f_{i+1}(nP), f_{i+1}((n+1)P)$. Then from the duplication formula, we could directly find $f_i(2nP), f_i(2(n+1)P), f_i((2n+1)P)$.

The doublings only require the points $f_i(0_E)$ which are given by the two curves E and E'. However the differential addition needs $f_i(P)$. So what we can do is compute $f_i(P), f_i(0_E)$ then apply our duplication formula. This inverse the order: rather than doing two isogeny images and two duplication at each step, we compute all the images first and then do all the duplications. We gain because the images $f_i(0_E)$ are free. We could expect to gain $2S + 1m_0$, but because our points $f_i(P)$ are no longer normalised, we only gain $2S + 1m_0 - M$ compared to the normal ladder with a normalised P.

In summary: we do a precomputation phase with all the $f_i(P)$. This cost $2S + 1m_0$ by bit, along with 2 field coefficients. Then we do our duplication formula: this cost $2S + 1m_0$ for our doublings, and $4M$ for our differential additions (again, because the $f_i(P)$ are not normalised). The final cost including the precomputation is $4M + 4S + 2m_0$. Further multiplication with the same base point P will cost $4M + 2S + 1m_0$. We note that this cost is the same whether P is normalised or not (because even if P is normalised, the $f_i(P)$ won't be).

When we know in advance P will be used (for public key encryption, or the first phase of DH key exchange), it is worth it to normalise the $f_i(P)$ at the cost of $1I$ by bit (the storage is then 1 coeff by bit). Then scalar multiplication will cost $3M + 2S + 1m_0$.

The big advantage compared to other time/memory trade off with elliptic curves (naf, window, …) is that the scalar multiplication is still a ladder with a double and diff add by bit, hence much less susceptible to side channel attack.

The same principle apply to the twisted theta model $\theta t w'$, by using the linear change of variable from the theta model, but we need some careful translation by $f_i(T_2)$ to gain $1M$ at each step (essentially we use a trick similar to the hybrid ladder): for the differential addition we assume that we have $f_{i+1}(nP), f_{i+1}((n+1)P + T_2)$ (say) and we compute $f_i((2n+1)P + T_2)$. (Doublings are no problem). We obtain the same cost as in the θ model,

except the initial translation by the two torsion point; likewise in the Montgomery cap Legendre model.

The formula are as follow (pending typos, I recommend to look at the code in [Rob23e] instead to be sure to use correct formulas...): given $(x_{Pi} : z_{Pi})$, the isogenous point P_{i+1} is given by: $X = (x_{Pi}^2 + z_{Pi}^2)b_i^2 : (x_{Pi}^2 - z_{Pi}^2)a_i^2)$. From P_{i+1} we can compute $2P_i$ via the dual isogeny: $(X+Z)^2 b_{i+1}, (X-Z)^2 a_{i+1})$. The more interesting part is the differential addition, given $P_{i+1} = (xg_P : zg_P), Q_{i+1} + T_{2_{i+1}} = (xg_{Q'} : zg_{Q'}), (P-Q)_i = (x_{PQ} : z_{PQ})$ we recover $(P+Q)_i$ via: $s = (xg_P + zg_P)(xg_{Q'} + zg_{Q'}); t = (xg_P - zg_P)(xg_{Q'} - zg_{Q'})$; $u = s+t; v = s-t; X = u/(x_{PQ}+z_{PQ}); Z = v/(x_{PQ}-z_{PQ}); (P+Q)_i = (X+Z : X-Z)$.

For Curve25519, since the two torsion is not rational, we need to move via a 2-isogeny to the curve above it which is both Montgomery and has full rational two torsion. Unfortunately the constant is large, so the cost of $4M + 4S + 2m_0$ when including the precomputation is essentially the same as with a standard Montgomery ladder: $5M + 4S + 1m_0$ (assuming P is normalised; we gain $1M$ on a non normalised point). Still, with the normalised precomputation, the cost of $3M + 2S + 1m_0$ is still very interesting, even with a large m_0.

The reason we work to work on the Montgomery cap Legendre model, is that if we want the relations $x(P+Q)z(P+Q), x(P-Q)z(P-Q)$ to factor through the isogeny f with kernel a 2-torsion point T, we need T to be of Montgomery type (equivalently the Tate pairing $e(T,T) = 1$, or the symmetric element in the theta group above T is rational). So the curve needs to be Montgomery, but the isogeneous curve should be too (because we go back and forth between the two curves), which is equivalent to the starting curve being in Legendre form.

6.3. **A general framework to find differential additions.** We can extend our general isogeny framework from Section 5.2 to differential additions. In this case we study the isogeny $\xi : (P_1, P_2) \to (P_1+P_2, P_1-P_2)$, the pullback $\xi^*(2(0_{E'}) \star 2(0_{E'}) = 4(0_E) \star 4(0_E)$. The kernel is given by the diagonal embedding of $E[2]$, and the symmetric lift giving our divisor descent is given by $h_T \otimes h_T$ for $T \in E[2]$ and h_T any of the two symmetric lift above T; even if h_T is not rational the tensor product is. We can thus compute the actions on $\Gamma(2(0_E) \star 2(0_E))^{\otimes 2}$, these span the even elements of $\Gamma(4(0_E) \star 4(0_E))$, of dimension 9. It follows that on $E' \times E'$ we can only construct the even elements in $\Gamma(2(0_E) \star 2(0_E))^+$, where the involution is given here by the descent of $(P_1, P_2) \to (P_1, -P_2)$, in other words we can only express elements invariant under the involution $P_1 + P_2 \mapsto P_1 - P_2$: $x(P_1+P_2)x(P_1-P_2)$, $z(P_1+P_2)z(P_1-P_2), x(P_1+P_2)z(P_1-P_2) + z(P_1+P_2)x(P_1-P_2)$.

Now, it is useful to factorize doubling through the isogeny f with kernel T: $[2] = \tilde{f} \circ f$, and we want to do the same with differential additions. In other words, we have $f(P_1), f(P_2)$ and we want to find from this some functions involving $P_1 + P_2, P_1 - P_2$. So consider $F : E \times E \to E' \times E'$ given by the diagonal of f. Then $\ker F = \{(0,0), (0,T), (T,0), (T,T)\}$; notice that it is not included in $\ker \xi$, so ξ does not factorize through F. So we need to consider the pushforward C of F and ξ of kernel $\mathrm{Ker}\, F + \mathrm{Ker}\, \xi$. From $f(P_1), f(P_2)$, we can not express all even functions in $P_1 + P_2, P_1 - P_2$, but we can only obtain those that descend to C, ie are invariants by $(0,T), (T,0)$ (or more precisely $1 \otimes h_T$ and $h_T \otimes 1$). It is instructive to look at the case $\lambda_T = 1$, in which case $x(P_1+P_2)x(P_1-P_2), z(P_1+P_2)z(P_1-P_2)$ are invariants. This explain the form the differential addition formula take for a Montgomery point.

As explained in Section 6.2, when doing a Montgomery ladder, we can then use a cycle of f and $\hat{f}$ to interleave the order of isogenies and differential additions: rather than computing images and differential additions (or doublings) at each step, we can compute iterated image, and then compute iterated differential additions (and doublings). The advantage is that in the

standard ladder we compute two images at each step, while here we only need to compute the iterated image of our base point, so we gain one image. On the other hand we need memory to store our iterated images, and we cannot assume that these images are normalised (unless we do a normalisation step on our points at the end). So to compute $N.P$ we compute $\log N$ iterated images of P (one by bit), then we do a ladder doing one doubling-through-isogeny, differential-addition through isogeny by step. This gives a general time/memory trade off for the Montgomery ladder.

Since we need to use differential addition formula that factorize through both f and $\tilde{f}$, the best case is thus when both are given by kernels of Montgomery type, ie our starting curve is in Montgomery cap Legendre.

7. Pairings

On a Kummer line, it is useful to interpret pairings as coming from the biextension law [Gro72; Sta08] associated to the divisor $2(0_E)$. It is shown in [Gro72] how the biextension gives rise to the Weil pairing, and [Sta08] extends this to the Tate pairing.

In this section I only give a very brief overview of the algorithm, and refer to the talk [Rob23b] for a bit more details.

For the biextension X associated to the divisor (0_E), an element $g_{P,Q}$ corresponds to a function on $k(E)$ with divisor $(P)+(Q)-(P+Q)-(0_E)$. The biextension partial group laws are given by:

$$(g_{P_1,Q} \star_1 g_{P_2,Q})(R) = g_{P_1,Q}(R) g_{P_2,Q}(R-P_1)$$
$$(g_{P,Q_1} \star_2 g_{P,Q_2})(R) = g_{P,Q_1}(R) g_{P,Q_2}(R) \frac{g_{Q_1,Q_2}(R-P)}{g_{Q_1,Q_2}(R)}$$

Moreover, since the divisor is symmetric, the biextension is symmetric too: $g_{P_1,Q} \star_1 g_{P_2,Q} = g_{Q,P_1} \star_2 g_{Q,P_2}$. This implies: $\mu_{P_1,P_2}(-P_3) = \mu_{P_2,P_3}(-P_1) = \mu_{P_3,P_1}(-P_2)$. A convenient way to represent a biextension element $g_{P,Q}$ is via (P,Q) and its evaluation on some point R. The group law becomes

$$(Q,P_1,c_1) \star_2 (Q,P_2,c_2) = c_1c_2 \frac{g_{P_1,P_2}(R-Q)}{g_{P_1,P_2}(R)},$$

and in particular we have:

$$g_{Q,P}^{\star_2,\ell} = g_{Q,P}(R)^\ell f_{\ell,P}((R-Q)-(R)),$$

where $\operatorname{div} f_{\ell,P} = \ell P - (\ell P) - (\ell-1)(0_E)$. Thus Miller's algorithm is simply the biextension exponentiation via this representation (and taking $R = 0_E$). We also have the following variant, using the symmetry:

$$(Q,P_1,c_1) \star_2 (Q,P_2,c_2) = c_1c_2\mu_{P_1,P_2}(-Q) = c_1c_2\mu_{P_1,Q}(-P_2),$$

which give the following alternative formula for the Miller addition:

$$f_{m+1,P}(-Q) = f_{m,P}(-Q)\mu_{mP,P}(-Q) = f_{m,P}(-Q)\mu_{P,Q}(-mP).$$

In particular, the biextension arithmetic gives the Tate and Weil pairing. Let $g_{P,Q} \in X(\mathbb{F}_q)$ be any element above (P,Q), $P \in E[\ell]$. since $\ell P = 0$, $g_{P,Q}^{\star_1,\ell}$ is a constant λ_P. If $\mu \in \mathbb{G}_m(\mathbb{F}_q)$ and $g'_{P,Q} = \mu \cdot g_{P,Q}$, then $g'^{\star_1,\ell}_{P,Q} = \mu^\ell \lambda_P$, and the class of λ_P in $\mathbb{F}_q^*/\mathbb{F}_q^{*,\ell}$ is the non reduced Tate pairing. Furthermore, $g_{P,Q}^{\star_1,q-1} = \lambda_P^{(q-1)/\ell}$ is the reduced Tate pairing $e_{T,\ell}(P,Q)$; it does not depends on the choice of $g_{P,Q}$. If $Q \in E[\ell]$, $g_{P,Q}^{\star_2,\ell} = \lambda_Q$; the Weil pairing is given by

$e_{W,\ell}(P,Q) = \lambda_P/\lambda_Q$. We also have similar formulas for the Ate and optimal Ate pairing, see [Rob23b].

We have the following monodromy interpretation of the pairings. The non reduced Tate pairing is then given by $g^{\ell}_{P,Q}$, which can be computed from $\ell\widetilde{P}, \widetilde{\ell P + Q}$, which in turn can be computed from a three way affine Montgomery ladder: 1 affine doubling and 2 affine differential addition by step. Equivalently, the non reduced Tate pairing is given by comparing $g^{\ell+1}_{P,Q}$ with $g_{P,Q}$, they differ by a projective factor λ_Q which is precisely the pairing. This λ_Q can be interpreted as a monodromy action: $Q \mapsto (\ell+1)Q$ is trivial at the level of the elliptic curve, but not at the biextension level. Likewise, the Weil pairing is given by the quotient of monodromy λ_Q/λ_P. The reduced Tate pairing is given by comparing $g^{q}_{P,Q}$ with $g_{P,Q}$, since $q-1$ is divisible by ℓ in our pairing situations, this is indeed the same as raising the non reduced Tate pairing to the power $(q-1)/\ell$. From this point of view the reduced Tate pairing is the Weil-Cartier pairing associated to $\pi_q - 1$.

For our efficient formulae, rather than using the Miller representation of the biextension elements, we will use the cubical torsor structure. We refer to [Bre83; Mor85] for cubical torsors.

We can indeed reinterpret the biextension law as follow: the key point is that with a symmetric line bundle, there is a *canonical* isomorphism $t_P^*L \otimes t_Q^*L \otimes t_R^*L \otimes t_S^*L \simeq t_U^*L \otimes t_V^*L \otimes t_W^*L \otimes t_X^*L$ whenever $P+Q+R+S = 2Z, U = Z-P, V = Z-Q, W = Z-R, X = Z-S$.

Specialising, we get partial group law on trivialisations of line bundle: $\tilde{0}, \widetilde{P}, \widetilde{Q}, \widetilde{P-Q} \mapsto \widetilde{P+Q}, \tilde{0}, \widetilde{P}, \widetilde{Q}, \widetilde{R}, \widetilde{P+Q}, \widetilde{P+R}, \widetilde{Q+R} \mapsto P + \widetilde{Q} + R$. Technically, these relations give the cubical torsor structure, which is a refinement of the arithmetic in the biextension.

(Note: in [Sta08] the biextension appears in the guise of elliptic nets. From our point of view, we can reinterpret elliptic nets as trivialisation of the line bundle $D = (0_E)$ at points P, notably by specifying the value of $Z(P)$ where Z is the section of (0_E). A slight difficulty is that Z has a zero on 0_E, so we need some offset to compute the pairings. The remarkable thing about elliptic nets is that even through we are on level 1 we can still compute the arithmetic of biextension through the linear recurrence of elliptic nets, see [Sta08] for details.

In [LR10; LR15], the biextension is hidden through the guise of the analytic Riemann relations giving the transcendental group law.)

We then represent an element $g_{P,Q}$ of the biextension by the trivialisations $\tilde{x}, \widetilde{x+P}, \widetilde{x+Q}, \widetilde{x+P+Q}$. Changing the trivialisations by $\lambda_x, \lambda_P, \lambda_Q, \lambda_{P+Q}$ give the same element iff $\lambda_x\lambda_{P+Q} = \lambda_P\lambda_Q$ (So our affine lifts represent a cubical torsor structure, and the associated biextension element is an equivalence class under this action).

The affine doublings and affine differential additions are formula lifting the standard projective doublings and projective differential additions. When working on the biextension we have more leeway, but when working on the cubical torsor structure we must be careful to use the correct affine formulas. We refer to the implementation in [Rob23e] for the explicit formula.

In the theta or twisted theta model, using [LR10; LR15] this amount to $7S + 7M + 2m_0$ by bit, assuming our base points are normalised (else add $2M$ by bit). On the Montgomery model, the biextension ladder costs $8M + 6S + 1m_0$ by bit.

By comparison, the best formula I have found for generic pairing computations in the Jacobian model cost $10M + 9S$ for doubling, and $11.5M + 3S$ by addition [BELL10].

In certain cases, we can compute the biextension exponentiation faster:

- In the general case, we compute $g^{\ell}_{P,Q}$ via one affine doubling and two affine differential additions by bits, for a total cost of $8M + 6S + 1m_0$ in the Montgomery model;
- For a self pairing, $P = Q$, we only need one affine doubling and one affine differential addition, for a total cost of $5M + 4S + 1m_0$ by bits. (A word of warning: for a fast self pairing we really need to use the cubical arithmetic rather than just the biextension arithmetic).
- When $n = 2^m$ is a power of 2, we also need only one affine doubling and one affine differential addition, for a total cost of $5M + 4S + 1m_0$ by bits.
- When $n = 2^m$ and $P = Q$, we only need one affine doubling, for a total cost of $2M + 2S + 1m_0$ by bits.

We can also do a standard exponentiation on $g_{P,Q}$ on our biextension, this allows to use the standard NAF and windowing method. We can do additions on the biextension model (at least with our representation), even through we are on the Kummer line on the underlying curve!

I worked out the formula in the theta model, using [LR15; LR16]: doubling cost 1 double and 1 diff add on the underlying curve, for a cost of $4M + 5S + 2m_0$. Addition is more complicated: on the underlying curve this amount to one (projective) compatible addition which cost $27M$ (I am not distinguishing M, S and m_0 here), followed by an affine three way addition which cost $17M$, for a grand total of $44M$. But since our base points are always the same (the ones we computed for our window), we can do some precomputations for these steps, and the compatible addition then cost $17M$, and the three way addition $13M$, for a total of $30M$.

Since doubling is $11M$, this might be competitive with the ladder method (which costs $16M$ by bit) when using a NAF-window with $w \geq 5$.

Remark 7.1. When working on the Kummer line, we are naturally working with the biextension associated to the divisor $2(0_E)$ rather than (0_E), because our coordinates $X, Z \in \Gamma(2(0_E))$. The corresponding biextension monodromy gives thus the square of the usual Tate and Weil pairing; which is no problem when ℓ is odd. This however lose one bit of information when ℓ is even; luckily in this case we can use the natural action of the theta group $G(2(0_E))$ on $\Gamma(2(0_E))$ to recover the Weil and Tate pairings exactly rather than just their squares. Once again we refer to [Rob23e] for the formulas.

7.1. **The Tate pairing for pairing based cryptogrpahy.** For pairing based cryptography on elliptic curves, it is convenient to use the Tate pairing with $P \in \mathbb{G}_1 \subset E(\mathbb{F}_q)$, $Q \in \mathbb{G}_2 \subset E(\mathbb{F}_{q^k})$, and k even to allow for denominator elimination.

Counting only operations involving the big field $\mathbb{F}_{q^k}$, Miller's algorithm cost $1M+1S+1m$ by doubling, and $1M+1m$ by addition. Here $1m$ denotes a multiplication between a coefficient in $\mathbb{F}_q$ and a coefficient in $\mathbb{F}_{q^k}$.

When denominator elimination is not possible (because k is odd or Q is not in $\mathbb{G}_2$), the cost becomes $2M + 2S + 1m$ by doubling, and $2M + 1m$ by addition.

Using our arithmetic of biextension on Kummer lines, only counting the operations on the big field, we have $2S + 1M + 2m$ by bit. So better than Miller's algorithm, except when denominator elimination is available.

7.2. **Monodromy leak.** It is well known, when $\mu_\ell \subset \mathbb{F}_q$, that the Tate pairing allows to reduce the DLP from an elliptic curve to $\mathbb{F}_q^*$.

From the point of view of étale torsors [Rob23g], the Tate pairing $e_{T,\ell}(P,Q)$ is an isomorphism class of the torsion $f^{-1}(Q)$ where f is the isogeny of kernel $\langle P\rangle$.

When $\mathbb{F}_q$ does not contains μ_ℓ (say ℓ prime for simplicity), a torsor is always trivial, ie $f^{-1}(Q)$ always contains one rational point.

However, we can still recover informations on the DLP if we manage to track explicit isomorphisms between $f^{-1}(Q)$ and $f^{-1}(n.Q)$. The theta group is precisely calibrated to keep track of such isomorphisms. Theta groups and biextensions are closely related (we will explore this topic further in [Rob23d]), and in this section we explore how to use biextensions to attack the DLP. (Very similar ideas were already pursued via elliptic nets in [LS08].)

We assume from now on that we are in this case.

The general idea is as follows: the biextension arithmetic is a juxtaposition of arithmetic on the underlying elliptic curve and in $\mathbb{F}_q^*$. When computing an exponentiation $n \mapsto n.P$, leaking instead a biextension exponentiation $n \mapsto g_P^n$ allows to recover n via a DLP in $\mathbb{F}_q^*$.

It might seem hard to leak such a biextension exponentiation on purpose, but from the pairing formula we see that since we are naturally working on affine coordinates, and the natural affine additions formulas are the ones coming from the biextension arithmetic, we see that on the contrary doing a Montgomery ladder leaks the biextension exponentiation as long as we don't randomize the coordinates (X,Z) by a factor $(\lambda X, \lambda Z)$ or we don't output the division $x = X/Z$.

There are two versions of the projective coordinates leak. The key idea is as follow: from our assumptions there is a unique lift $\widetilde{g_P}$ of P in the biextension that is still of order ℓ. This "canonical lift" can be computed efficiently by a scalar multiplication in the biextension, this scalar being determined by being 0 modulo $p-1$ and 1 modulo ℓ.

We now start with $P = (x_P, 1)$ corresponding to some $g_P = \lambda_1 \widetilde{g_P}$, and with overwhelming probability λ_1 is not trivial (we use g_P as a shortcut for the biextension element associated to $g_{P,P}$). The value of g_P^n is leaked, which gives us $g_P^n = \lambda_2 \widetilde{g_{nP}}$. But since $\widetilde{g_{nP}} = \widetilde{g_P}^n$, we get that $\lambda_2 = \lambda_1^n$. From λ_2, λ_1, we recover n via a DLP in $\mathbb{F}_q^*$. This version requires g_P^n, so a leak of both $\widetilde{nP}$, $\widetilde{(n+1)P}$. Furthermore, the biextension arithmetic is slightly different from the way the Montgomery ladder is implemented in practice, so we need to do some slight adjustments (see below).

The stronger version of the projective coordinate leak only requires a leak of $\widetilde{nP}$. This time we need the full power of the cubical torsor structure rather than just of biextension; there is still a unique (using our assumption that ℓ is prime to $q-1$) canonical lift $\widetilde{P}$ which is of ℓ-torsion and can be efficiently computed from P. So we start with $P = (x_P, 1) = \lambda_1 \widetilde{P}$ and we are leaked $n.P = (X,Z) = \lambda_2 \widetilde{nP}$. This time, we have $\lambda_2 = \lambda_1^{n^2}$, so we need a DLP and then solve a square root.

However, taking into account that the actual Montgomery ladder is different from the exact cubical torsor structure arithmetic, we need to correct by some factor, so we actually solve a more general degree two polynomial.

Explicitly, computing $n.P$ via the standard ladder arithmetic rather than via the correct cubical ladder, we are off by a factor $(4x_P)^{n(2^b-n)}$ where b is the bit length of n. Taking a multiplicative generator ζ of F_q^*, we thus need to solve the equation:

$$X^2(\mathrm{dlp}_\zeta(\lambda_1) - \mathrm{dlp}_\zeta(4x_P)) + X 2^b \,\mathrm{dlp}_\zeta(4x_P) - \mathrm{dlp}_\zeta(\lambda_2) = 0. \tag{16}$$

The number of solutions depends to check for afterwards depends on the number of prime factors of $q-1$. In good cases, there are few enough factors to reconstruct the solutions modulo a large enough modulus efficiently. We refer to the code [Rob23e] for more details.

We call this a monodromy leak for the following reason. We'll use the biextension version rather than the cubical version for simplicity. Assume that we are not given the projective coordinate leak of $n.P$, which encodes the information about g_P^n. We can still take any biextension element $g_{P,nP}$ above nP. There is a m such that $g_{P,nP} = g_P^m$, where the value of m is determined modulo $\ell(q-1)$ and is congruent to n modulo ℓ. We have $g_{P,nP} = \lambda_2 \widetilde{g_{P,nP}}$ and $g_P = \lambda_1 \widetilde{g_P}$, so we have the equation $\lambda_2 = \lambda_1^m$ in $\mathbb{F}_q^*$. Unfortunately, this only recover the value of m modulo $q-1$ (at best, if λ_1 is a multiplicative generator), which gives no information on n modulo ℓ since ℓ is prime to $q-1$. The reason the projective coordinate leak above works is that in this case we know that $m \leq \ell$, so is equal to n. Essentially, we know that the value of $g_{P,nP}$ we obtain from the projective coordinate leak is g_P^m with m small enough and not wrapping an unknown number of time around ℓ; which is why we call it a monodromy leak.

Compared to [NSS04], our monodromy leak requires to know the starting coordinates (X_P, Z_P) used in the ladder (usually the point P is normalised so that $z_P = 1$ which is the assumption we have used in the above formulaes; but the general case is not harder, as long as we know the choice of z_P), rather than just the leak of the projective coordinates of $nP = (X_{nP}, Z_{nP})$. On the other hand, it is much more devastating: rather than leaking a few bits of n, we recover it fully via some DLPs in $\mathbb{F}_q^*$, so in subexponential time. (In practice q is of 256 bits, so the DLP is quite effective).

The monodromy leak is thwarted e.g. by doing a constant time division at the end to only send $x_{nP} = X_{nP}/Z_{nP}$. For extra security measure, a supplementary countermeasure is also to mask the projective coordinates of P by a random scalar at the beginning. This protect in case side channels information allows to recover some informations on the intermediate projective coordinates during the ladder. This means that P won't be normalised any longer, so this adds $1M$ by bits in the usual ladder, but luckily the complexity of the time/memory trade off described in Section 6 does not depends on whether P is normalised or not.

Remark 7.2. Fre Vercauteren informed me that the curve NISTp521 uses the prime $p = 2^{521} - 1$ such that $\mathbb{F}_p^*$ has very smooth order (the largest factor has 60 bits). The DLP is very easy in this field.

It is plausible that the monodromy leak described here for the Montgomery ladder extends to more general scalar multiplication, albeit with a more complicated polynomial depending on the exact implementation of the scalar multiplication.

More precisely, what is certainly true is that the biextension and cubical torsor structure exist for all models (see the code [Rob23e]), and that we can efficiently compute "canonical lifts" as above. The only issue is that the scalar multiplication implemented, when interpreted on the affine lifts, won't be the same as the cubical multiplication. For the usual Montgomery ladder, it was easy to keep track of the corrective factor $(4x_P)^{n(2^b-n)}$ above, because the formulas are quite close to the "correct" cubical formulas. In general, it is plausible that there is still a corrective factor that can still be expressed in the exponent as some polynomial in n. This then would give a more complicated polynomial equation than Equation (16).

The main difficulty would be to handle the addition: the cubical arithmetic really needs to use some differential additions (or alternatively would write $(2n+1)P$ as a three way addition $(2n+1)P, P, nP, nP; 0, 2nP, (n+1)P, (n+1)P$), which Kummer lines arithmetic

also uses (maybe with a different constant than the cubical one) but not standard elliptic curve arithmetic.

Anyway, going further into wild speculations, it would not be surprising if the NSA was aware of these kind of "monodromy attacks" or variants. Continuing our wild speculations, we remark that the NIST curves are from 1999, and at that time a 512 bits DLP in $\mathbb{F}_p^*$ was probably quite expensive even for the NSA: even in 2005 the public record for a DLP was for 430 bits. But selecting p such that $p-1$ is smooth would render the DLP in $\mathbb{F}_p^*$ trivial, and at that time [NSS04] was not yet published, so probably not all implementations were protected against projective coordinates leaks…

References

[AGB20] A. C. Aldaya, C. P. García, and B. B. Brumley. "From A to Z: Projective coordinates leakage in the wild". In: *IACR Transactions on Cryptographic Hardware and Embedded Systems* (2020), pp. 428–453 (cit. on p. 4).

[BRS23] R. Barbulescu, D. Robert, and N. Sarkis. "Models of Kummer lines and Galois representations". June 2023. In preparation. (Cit. on pp. 2, 6, 9).

[BELL10] J. Boxall, N. El Mrabet, F. Laguillaumie, and D.-P. Le. "A variant of miller's formula and algorithm". In: *Pairing-Based Cryptography-Pairing 2010: 4th International Conference, Yamanaka Hot Spring, Japan, December 2010. Proceedings 4*. Springer. 2010, pp. 417–434 (cit. on pp. 3, 19).

[Bre83] L. Breen. *Fonctions thêta et théoreme du cube*. Vol. 980. Springer, 1983 (cit. on p. 19).

[CH17] C. Costello and H. 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 23*. Springer. 2017, pp. 303–329 (cit. on pp. 13, 14).

[CLN16] C. Costello, P. Longa, and M. Naehrig. "Efficient algorithms for supersingular isogeny Diffie-Hellman". In: *Advances in Cryptology–CRYPTO 2016: 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I 36*. Springer. 2016, pp. 572–601 (cit. on pp. 3, 13).

[DJP14] L. De Feo, D. Jao, and J. Plût. "Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies". In: *Journal of Mathematical Cryptology* 8.3 (2014), pp. 209–247 (cit. on p. 11).

[Gro72] A. Grothendieck. *Groupes de Monodromie en Géométrie Algébrique: SGA 7*. Springer-Verlag, 1972 (cit. on p. 18).

[HR19] H. Hisil and J. Renes. "On kummer lines with full rational 2-torsion and their usage in cryptography". In: *ACM Transactions on Mathematical Software (TOMS)* 45.4 (2019), pp. 1–17 (cit. on p. 6).

[LS08] K. E. Lauter and K. E. Stange. "The elliptic curve discrete logarithm problem and equivalent hard problems for elliptic divisibility sequences". In: *International Workshop on Selected Areas in Cryptography*. Springer. 2008, pp. 309–327 (cit. on p. 21).

[LR10] D. Lubicz and D. Robert. "Efficient pairing computation with theta functions". In: ed. by G. Hanrot, F. Morain, and E. Thomé. Vol. 6197. Lecture Notes in Comput. Sci. 9th International Symposium, Nancy, France, ANTS-IX, July 19-23, 2010, Proceedings. Springer–Verlag, July 2010. DOI: 10.1007/978-

3-642-14518-6_21. URL: http://www.normalesup.org/~robert/pro/publications/articles/pairings.pdf. Slides: 2010-07-ANTS-Nancy.pdf (30min, International Algorithmic Number Theory Symposium (ANTS-IX), July 2010, Nancy), HAL: hal-00528944. (Cit. on p. 19).

[LR15] D. Lubicz and D. Robert. "A generalisation of Miller's algorithm and applications to pairing computations on abelian varieties". In: *Journal of Symbolic Computation* 67 (Mar. 2015), pp. 68–92. DOI: 10.1016/j.jsc.2014.08.001. URL: http://www.normalesup.org/~robert/pro/publications/articles/optimal.pdf. HAL: hal-00806923, eprint: 2013/192. (Cit. on pp. 19, 20).

[LR16] D. Lubicz and D. Robert. "Arithmetic on Abelian and Kummer Varieties". In: *Finite Fields and Their Applications* 39 (May 2016), pp. 130–158. DOI: 10.1016/j.ffa.2016.01.009. URL: http://www.normalesup.org/~robert/pro/publications/articles/arithmetic.pdf. HAL: hal-01057467, eprint: 2014/493. (Cit. on p. 20).

[Mor85] L. Moret-Bailly. *Pinceaux de variétés abéliennes*. Société mathématique de France, 1985 (cit. on p. 19).

[NSS04] D. Naccache, N. P. Smart, and J. Stern. "Projective coordinates leak". In: *Advances in Cryptology-EUROCRYPT 2004: International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004. Proceedings 23*. Springer. 2004, pp. 257–267 (cit. on pp. 2, 4, 22, 23).

[OLHFR18] T. Oliveira, J. López, H. Hışıl, A. Faz-Hernández, and F. Rodríguez-Henríquez. "How to (pre-) compute a ladder: Improving the performance of X25519 and X448". In: *Selected Areas in Cryptography–SAC 2017: 24th International Conference, Ottawa, ON, Canada, August 16-18, 2017, Revised Selected Papers 24*. Springer. 2018, pp. 172–191 (cit. on pp. 14–16).

[Rei23] K. Reijnders. "Effective Pairings in Isogeny-based Cryptography". In: *Cryptology ePrint Archive* (2023) (cit. on p. 3).

[Ren18] J. Renes. "Computing isogenies between Montgomery curves using the action of (0, 0)". In: *Post-Quantum Cryptography: 9th International Conference, PQCrypto 2018, Fort Lauderdale, FL, USA, April 9-11, 2018, Proceedings*. Springer. 2018, pp. 229–247 (cit. on pp. 9, 13).

[Rob22] D. Robert. "Arithmetic on Kummer lines". Oct. 2022. In preparation. (Cit. on p. 2).

[Rob23a] D. Robert. "A note on optimising 2^n-isogenies in higher dimension". June 2023. URL: http://www.normalesup.org/~robert/pro/publications/notes/2023-06-optimising_isogenies.pdf (cit. on pp. 5, 6, 8, 9, 16).

[Rob23b] D. Robert. "Arithmetic and pairings on Kummer lines". Leuven isogeny days 4, Leuven. Oct. 2023. URL: http://www.normalesup.org/~robert/pro/publications/slides/2023-10-Leuven.pdf (cit. on pp. 18, 19).

[Rob23c] D. Robert. "Biextensions and Pairings on Kummer lines". Aug. 2023. In preparation. (Cit. on p. 2).

[Rob23d] D. Robert. "Canonical liftings to biextensions and theta groups". Aug. 2023. In preparation. (Cit. on p. 21).

[Rob23e] D. Robert. "Kummer Line". Toolbox for computing on Kummer lines. Oct. 2023. URL: https://gitlab.inria.fr/roberdam/kummer-line (cit. on pp. 2, 17, 19, 20, 22).

[Rob23f] D. Robert. "Projective coordinate leaks revisited". Oct. 2023. In preparation. (Cit. on p. 2).

[Rob23g] D. Robert. "The geometric interpretation of the Tate pairing and its applications". Feb. 2023. URL: http://www.normalesup.org/~robert/pro/publications/articles/geometric_tate_pairing.pdf. eprint: 2023/177, HAL: hal-04295743v1. (Cit. on p. 21).

[RS24] D. Robert and N. Sarkis. "Computing 2-isogenies between Kummer lines". Jan. 2024. URL: http://www.normalesup.org/~robert/pro/publications/articles/kummer_isogenies.pdf. eprint: 2024/037. (Cit. on p. 2).

[Sta08] K. Stange. "Elliptic nets and elliptic curves". PhD thesis. Brown University, 2008. URL: https://repository.library.brown.edu/studio/item/bdr:309/PDF/ (cit. on pp. 18, 19).

INRIA Bordeaux–Sud-Ouest, 200 avenue de la Vieille Tour, 33405 Talence Cedex FRANCE
Email address: damien.robert@inria.fr
URL: http://www.normalesup.org/~robert/

Institut de Mathématiques de Bordeaux, 351 cours de la liberation, 33405 Talence cedex FRANCE

[BCR10] G. Bisson, R. Cosset, and D. Robert. *AVIsogenies*. Magma package devoted to the computation of isogenies between abelian varieties. 2010. URL: https://www.math.u-bordeaux.fr/~damienrobert/avisogenies/. Free software (LGPLv2+), registered to APP (reference IDDN.FR.001.440011.000.R.P.2010.000.10000). Latest version 0.7, released on 2021-03-13. (Cit. on pp. 7, 27).

[CR15] R. Cosset and D. Robert. "An algorithm for computing (ℓ, ℓ)-isogenies in polynomial time on Jacobians of hyperelliptic curves of genus 2". In: *Mathematics of Computation* 84.294 (Nov. 2015), pp. 1953–1975. DOI: 10.1090/S0025-5718-2014-02899-8. URL: http://www.normalesup.org/~robert/pro/publications/articles/niveau.pdf. HAL: hal-00578991, eprint: 2011/143. (Cit. on p. 27).

[DMPR23a] P. Dartois, L. Maino, G. Pope, and D. Robert. "An Algorithmic Approach to $(2, 2)$-isogenies in the Theta Model and Applications to Isogeny-based Cryptography". Nov. 2023. URL: http://www.normalesup.org/~robert/pro/publications/articles/_2_2__isogenies_in_the_theta_model.pdf. eprint: 2023/1747. (Cit. on pp. 27, 63).

[DMPR23b] P. Dartois, L. Maino, G. Pope, and D. Robert. *ThetaIsogenies*. Fast computations of isogenies in dimension two. Nov. 2023. URL: https://github.com/ThetaIsogenies/two-isogenies (cit. on p. 63).

[Dud16] A. Dudeanu. *Computational aspects of jacobians of hyperelliptic curves*. Tech. rep. EPFL, 2016 (cit. on p. 7).

[DJRV22] A. Dudeanu, D. Jetchev, D. Robert, and M. Vuille. "Cyclic Isogenies for Abelian Varieties with Real Multiplication". In: *Moscow Mathematical Journal* 22 (Feb. 2022), pp. 613–655. URL: http://www.normalesup.org/~robert/pro/publications/articles/cyclic.pdf. HAL: hal-01629829. (Cit. on pp. 7, 27).

[FLR11] J.-C. Faugère, D. Lubicz, and D. Robert. "Computing modular correspondences for abelian varieties". In: *Journal of Algebra* 343.1 (Oct. 2011), pp. 248–277. DOI: 10.1016/j.jalgebra.2011.06.031. arXiv: 0910.4668 [cs.SC]. URL: http://www.normalesup.org/~robert/pro/publications/articles/modular.pdf. HAL: hal-00426338. (Cit. on p. 27).

[Kan97] E. Kani. "The number of curves of genus two with elliptic differentials." In: *Journal für die reine und angewandte Mathematik* 485 (1997), pp. 93–122 (cit. on p. 55).

[KPR20] J. Kieffer, A. Page, and D. Robert. "Computing isogenies from modular equations between Jacobians of genus 2 curves". Oct. 2020. arXiv: 2001.04137 [math.AG]. URL: http://www.normalesup.org/~robert/pro/publications/articles/modular_isogenies_g2.pdf. HAL: hal-02436133. (Cit. on p. 27).

[LR12] D. Lubicz and D. Robert. "Computing isogenies between abelian varieties". In: *Compositio Mathematica* 148.5 (Sept. 2012), pp. 1483–1515. DOI: 10.1112/S0010437X12000243. arXiv: 1001.2016 [math.AG]. URL: http://www.normalesup.org/~robert/pro/publications/articles/isogenies.pdf. HAL: hal-00446062. (Cit. on p. 27).

[LR16] D. Lubicz and D. Robert. "Arithmetic on Abelian and Kummer Varieties". In: *Finite Fields and Their Applications* 39 (May 2016), pp. 130–158. DOI: 10.1016/j.ffa.2016.01.009. URL: http://www.normalesup.org/~robert/pro/publications/articles/arithmetic.pdf. HAL: hal-01057467, eprint: 2014/493. (Cit. on p. 39).

[LR22] D. Lubicz and D. Robert. "Fast change of level and applications to isogenies". In: *Research in Number Theory (ANTS XV Conference)* 9.1 (Dec. 2022). DOI: 10.1007/s40993-022-00407-9. URL: http://www.normalesup.org/~robert/pro/publications/articles/change_level.pdf. HAL: hal-03738315. (Cit. on p. 27).

[Rob21a] D. Robert. "Efficient algorithms for abelian varieties and their moduli spaces". HDR thesis. Université Bordeaux, June 2021. URL: http://www.normalesup.org/~robert/pro/publications/academic/hdr.pdf. Slides: 2021-06-HDR-Bordeaux.pdf (1h, Bordeaux). (Cit. on pp. 5, 27).

[Rob21b] D. Robert. *General theory of abelian varieties and their moduli spaces*. Mar. 2021. URL: http://www.normalesup.org/~robert/pro/publications/books/avtheory.pdf. Draft version. (Cit. on p. 5).

[Rob23] D. Robert. "Kummer Line". Toolbox for computing on Kummer lines. Oct. 2023. URL: https://gitlab.inria.fr/roberdam/kummer-line (cit. on p. 101).

[RS24] D. Robert and N. Sarkis. "Computing 2-isogenies between Kummer lines". Jan. 2024. URL: http://www.normalesup.org/~robert/pro/publications/articles/kummer_isogenies.pdf. eprint: 2024/037. (Cit. on p. 101).

[Vui20] M. L. Vuille. *Computing cyclic isogenies between principally polarized abelian varieties over finite fields*. Tech. rep. EPFL, 2020 (cit. on p. 7).