A generalisation of Miller's algorithm and applications to pairing computations on abelian varieties

David Lubicz, Damien Robert

A generalisation of Miller's algorithm and applications to pairing computations on abelian varieties

David Lubicz[1,2], Damien Robert[3]

[1] DGA-MI, BP 7419, F-35174 Bruz
[2] IRMAR, Universté de Rennes 1, Campus de Beaulieu, F-35042 Rennes
[3] Centre de Recherche Inria Bordeaux - Sud-Ouest
200, avenue de la Vieille Tour
33405 Talence cedex

Abstract. In this paper, we use the theory of theta functions to generalize to all abelian varieties the usual Miller's algorithm to compute a function associated to a principal divisor. We also explain how to use the Frobenius morphism on abelian varieties defined over a finite field in order to shorten the loop of the Weil and Tate pairings algorithms. This extend preceding results about ate and twisted ate pairings to all abelian varieties. Then building upon the two preceding ingredients, we obtain a variant of optimal pairings on abelian varieties. Finally, by introducing new addition formulas, we explain how to compute optimal pairings on Kummer varieties. We compare in term of performance the resulting algorithms to the algorithms already known in the genus one and two case.

1 Introduction

The computation of Weil and Tate pairings has important applications in arithmetic and cryptography. Almost all the known algorithms to compute these pairings on elliptic curves rely on Miller's algorithm [21] to evaluate at a certain point a function associated to a principal divisor.

The improvements over the initial version of Miller's algorithm have followed two main approaches:

- making the basic loop of Miller's algorithm quicker with efficient arithmetic;
- reducing the number of loops of the algorithm by using endomorphisms of the curve.

For a curve defined over a finite field, using the absolute Frobenius endomorphism, this last line of ideas has led to the definition of eta-pairings [2], ate-pairings [14,12] and optimal pairings [28].

The paper [20] describes a new algorithm based on the theory of theta functions to compute Weil and Tate pairings which apply to all abelian varieties.

It is a natural question to ask whether the known optimizations of the classical Miller's algorithm on elliptic curves can be adapted to the algorithms presented in [20]. In this paper, we focus on the optimisations which consist in reducing the number of loops in pairing computation algorithms by using non trivial endomorphisms of the abelian varieties. We won't deal here with the conversion between Weierstrass coordinates of an elliptic curve or Mumford coordinates of a Jacobian of an hyperelliptic curve and theta functions. These conversion formulas come from the well known Thomae's formula, and are described in more details in [11,10,30,5].

Classical pairing computation algorithms rely on Miller's algorithm to compute the function f of a genus g curve C defined up to a constant factor by a divisor $D \in \mathrm{Pic}^0(C)$ linearly equivalent to 0. For $D_a, D_b \in \mathrm{Pic}^0(C)$, denote by f_{D_a,D_b} the function given (up to a constant factor) by the divisor $D_a + D_b - (D_a \oplus D_b)$ where $D_a \oplus D_b$ is the reduced divisor associated to $D_a + D_b$. Miller's algorithm is based on a double and add loop that iterates a formula which gives the function f_{D_a,D_b} . We note that such a formula is specific to a model of a curve. Although in the case of the Weierstrass model of an elliptic curve it is immediately provided by the definition of the group law, for other models it may require some computations to obtain it (see for instance [1,8]). Using the classical Riemann relations for theta functions, we generalize Miller's algorithm to all abelian varieties. In our context, we compute the function defined up to a constant factor by a divisor D on an abelian variety with Chern class 0 linearly equivalent to 0. Our method rely on the projective embedding of a principally polarized abelian variety provided by a power of its theta divisor and therefore is not restricted to Jacobians of curves.

Once we know how to compute the Miller's functions, it is easy to apply the optimisations of the ate and optimal ate pairings to our setting. However, the formula that we obtain to compute the Weil and Tate pairings using the classical Miller's algorithm with theta functions is slower than the algorithms presented in [20], which use a different (and faster) subset of the Riemann relations that we call *differential additions*. Our second contribution is to extend the improvements of the ate and optimal ate pairings to this modified Miller's loop based on differential additions by studying the action of the absolute Frobenius endomorphism on points given by the theta coordinates.

To get even faster formulas, one can compute with a Kummer variety, which is the quotient of an abelian variety by the (-1) automorphism acting on it, by using theta functions of level 2 which are invariants by the action of (-1). There is no addition law on a Kummer variety, since we can not distinguish a point from its opposite. Still, there is an action of $\mathbb{N}$ on the points of a Kummer variety that one can compute with the differential additions. By carrying the pairing to this structure, we were led to introduce *symmetric pairings* in [20] by using differential additions on level 2 theta functions. Our third contribution is then to extend all the results previously discussed to this setting; this can be seen as a generalization of [9] to higher dimension. For this, we introduce new addition laws deduced from Riemann relations that can be used in the case of level 2

theta functions. Finally, we give an overview of the theoretical complexity of our algorithms.

In order to avoid too much formalism, we have chosen to present all the formulas of this paper using the classical analytic theory of theta functions. Nonetheless, it should be understood that most of our algorithms apply in general to abelian varieties defined over any field of characteristic not equal to 2. To see this one can invoke the Lefschetz's principle or use Mumford's theory of algebraic theta functions. There are part of the paper which are more specific to abelian varieties over finite field because we use the Frobenius endormophism in order to speed up the computations. In this case, we clearly state this hypothesis at the beginning of the section.

The paper is organised as follows: in Section 2, we recall some notations and well known results about theta functions. In Section 3, we describe different operations which can be computed on any abelian variety using Riemann relations. Section 4 is devoted to a generalisation of Miller's algorithm. In Section 5, we recall the standard definition of the Weil and Tate pairings on abelian varieties, and explain how to compute them using the results from the previous section. We give two ways of computing these pairings: the usual Miller's algorithm applied to theta functions and the utilisation of differential additions as in [20]. In Section 6, we explain how to extend this to compute the ate pairing on an abelian variety defined over a finite field, while in Section 7, the case of the optimal ate pairing is treated. Finally in Section 8, we give an overview of the complexity.

2 Some notations and basic facts

In this section, in order to fix the notations, we recall some well known facts about analytic theta functions (see for instance [25,15]). Let $\mathbb{H}_g$ be the g dimensional Siegel upper-half space which is the set of $g \times g$ symmetric matrices Ω whose imaginary part is positive definite. For $\Omega \in \mathbb{H}_g$, we denote by $\Lambda_\Omega = \Omega\mathbb{Z}^g + \mathbb{Z}^g$ the lattice of $\mathbb{C}^g$ defined by Ω. Any abelian variety $\mathscr{A}$ of dimension g over $\mathbb{C}$ with a principal polarisation is analytically isomorphic to $\mathbb{C}^g/\Lambda_\Omega$ for a certain $\Omega \in \mathbb{H}_g$. In the rest of this paper, we denote by $\pi : \mathbb{C}^g \to \mathbb{C}^g/\Lambda_\Omega = \mathscr{A}$ the canonical projection. The classical theory of theta functions gives a lot of functions on $\mathbb{C}^g$ that are pseudo-periodic with respect to Λ_Ω and can be used as a projective coordinate system for $\mathscr{A}$. More precisely, for $a, b \in \mathbb{Q}^g$, the theta function with rational characteristics (a, b) is an analytic function on $\mathbb{C}^g \times \mathbb{H}_g$ given by:

$$\theta\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right](z, \Omega) = \sum_{n \in \mathbb{Z}^g} \exp\left[\pi i^t (n+a).\Omega.(n+a) + 2\pi i^t (n+a).(z+b)\right]. \quad (1)$$

For all $m, n \in \mathbb{Z}^g$, we have:

$$\theta\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right](z+\Omega m+n) = \exp(2\pi i({}^t a.n - {}^t b.m) - \pi i^t m.\Omega.m - 2\pi i^t m.z)\theta\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right](z, \Omega). \quad (2)$$

We say that a function f on $\mathbb{C}^g$ is Λ_Ω-quasi-periodic of level $n \in \mathbb{N}$ if for all $z \in \mathbb{C}^g$ and $m \in \mathbb{Z}^g$, we have:$f(z+m) = f(z)$, $f(z+\Omega.m) = \exp(-\pi i n^t m.\Omega.m -$

$2\pi i n^t z.m) f(z)$. For any $n \in \mathbb{N}^*$, the set $H_{\Omega,n}$ of Λ_Ω-quasi-periodic functions of level n is a finite dimensional $\mathbb{C}$-vector space whose basis can be given by the theta functions with characteristics: $(\theta \left[\begin{smallmatrix} 0 \\ b/n \end{smallmatrix}\right](z, n^{-1}.\Omega))_{b\in[0,\dots,n-1]^g}$. If $n = k^2$, then an alternative basis of $H_{\Omega,n}$ is $(\theta \left[\begin{smallmatrix} a/k \\ b/k \end{smallmatrix}\right](kz, \Omega))_{a,b\in[0,\dots,k-1]^g}$. A theorem of Lefschetz tells that if $n \geq 3$, the functions in $H_{\Omega,n}$ give a projective embedding of $\mathscr{A}$ in $\mathbb{P}^{n^g-1}$, the projective space over $\mathbb{C}$ of dimension $n^g - 1$. For $n = 2$, the functions in $H_{\Omega,2}$ do not give a projective embedding of $\mathscr{A}$. Indeed, it is easy to check that for all $f \in H_{\Omega,2}$, we have $f(-z) = f(z)$. Under some well known general conditions [16, Corollary 4.5.2], the image of the embedding defined by $H_{\Omega,2}$ in $\mathbb{P}^{n^2-1}$ is the Kummer variety associated to $\mathscr{A}$, which is the quotient of $\mathscr{A}$ by the automorphism -1.

Once we have chosen a level $n \in \mathbb{N}$ and a matrix period Ω, for the rest of this paper, we adopt the following conventions: we let $Z(\overline{n}) = (\mathbb{Z}/n\mathbb{Z})^g$ and for a point $z_P \in \mathbb{C}^g$ and $i \in Z(\overline{n})$ we put $\theta_i(z_P) = \theta \left[\begin{smallmatrix} 0 \\ i/n \end{smallmatrix}\right](z_P, \Omega/n)$. For $\ell, n \in \mathbb{N}$, such that ℓ divides n we will implicitly consider $Z(\overline{\ell})$ as a subgroup of $Z(\overline{n})$ via the morphism $x \mapsto (n/\ell).x$.

We denote by Θ_n the theta divisor of level n on $\mathscr{A}$ which is the divisor of zero of $\theta \left[\begin{smallmatrix} 0 \\ 0 \end{smallmatrix}\right](z, n^{-1}.\Omega)$. There is an isogeny $\varphi_n : \mathscr{A} \to \hat{\mathscr{A}} = \mathrm{Pic}^0_{\mathscr{A}}$, defined by $x \mapsto \tau_x^* \Theta_n - \Theta_n$ where τ_x is the morphism of translation by x on $\mathscr{A}$. The kernel of φ_n is $\mathscr{A}[n]$. For $n = 1$ we let $\Theta_1 = \Theta$. If $\mathscr{L}_n$ is the line bundle corresponding to the polarization φ_n (or the divisor Θ_n), we have $\mathscr{L}_n = \mathscr{L}_1^n$. We denote by $K(\mathscr{A})$ the function field of $\mathscr{A}$ and if $f \in K(\mathscr{A})$, we denote by (f) the divisor of the function f. Let $Z^0(\mathscr{A})$ be the group of 0-cycles of $\mathscr{A}$ that is the free commutative group over the set of closed points of $\mathscr{A}$. If $D = \sum_{i\in I} n_i(P_i)$ is an element of $Z^0(\mathscr{A})$, we let $\mathrm{Supp}(D)$ be the reduced zero dimensional variety $\cup_{i\in I} P_i$. If $f \in K(\mathscr{A})$ has no poles nor zeroes on $\mathrm{Supp}(D)$, we put $f(D) = \prod_{i\in I} f(P_i)^{n_i}$.

We recall the following theorem from [20, Theorem 1] which is a version of the usual Riemann addition formula for theta functions:

Theorem 1. *Let $i, j, k, l \in Z(\overline{2n})$. We suppose that $i + j$, $i + k$ and $i + l \in Z(\overline{n})$. Let $\hat{Z}(\overline{2})$ be the dual group of $Z(\overline{2})$. For all $\chi \in \hat{Z}(\overline{2})$ and $z_1, z_2, z_3, z_4 \in \mathbb{C}^g$, let*

$$L_1 = \sum_{\eta\in Z(\overline{2})} \chi(\eta)\theta_{i+j+\eta}(z_1 + z_2)\theta_{i-j+\eta}(z_1 - z_2),$$
$$L_2 = \sum_{\eta\in Z(\overline{2})} \chi(\eta)\theta_{k+l+\eta}(z_3 + z_4)\theta_{k-l+\eta}(z_3 - z_4),$$
$$L_3 = \sum_{\eta\in Z(\overline{2})} \chi(\eta)\theta_{i+k+\eta}(z_1 + z_4)\theta_{i-k+\eta}(z_1 - z_4),$$
$$L_4 = \sum_{\eta\in Z(\overline{2})} \chi(\eta)\theta_{j+l+\eta}(z_3 + z_2)\theta_{j-l+\eta}(z_3 - z_2),$$

then we have

$$L_1.L_2 = L_3.L_4. \tag{3}$$

3 Addition laws deduced from Riemann relations

In this section, we explain how to compute certain operations on the set of geometric points of $\mathscr{A}$ using Riemann equations. These operations will be used in the algorithms of the next sections, but they may be interesting for other applications, for instance for the purpose of computing on Kummer varieties.

3.1 Normal additions

Let $\mathscr{A}$ be an abelian variety over $\mathbb{C}$ with period matrix Ω. For n a positive integer, we represent $\mathscr{A}$ as a closed subvariety of $\mathbb{P}^{Z(\overline{n})}$ by the way of level n theta functions and we suppose that 2 divides n. Such an embedding is uniquely determined once we have chosen a numbering of a basis of $H_{\Omega,n}$: by default we take $\{\theta_i,\, i \in Z(\overline{n})\}$. With this convention, by a theorem of Mumford, if $4|n$, the resulting embedding of $\mathscr{A}$ in $\mathbb{P}^{n^g-1}$ is defined by the equations of Theorem 1 by taking $z_2 = z_3 = z_4 = 0$.

Let K be a number field. In the rest of this paper, we suppose that this embedding is defined over K or, say in another way, that the projective point $0 = (\theta_i(0))_{i\in Z(\overline{n})}$ corresponding to the neutral element of $\mathscr{A}$ is defined over K.

Now if $4|n$, from the knowledge of $P = (P_i)_{i\in Z(\overline{n})}$ and $Q = (Q_i)_{i\in Z(\overline{n})}$, one can compute the (projective) point $P + Q = ((P + Q)_i)_{i\in Z(\overline{n})}$ using Theorem 1 with $z_3 = z_4 = 0$. We just have to check that the L_2 factor of equation (1) does not vanish too often which is a consequence of [20, Proposition 3]. We write $P + Q = \mathrm{NormalAdd}(P, Q)$.

We illustrate this with $n = 4$ and $g = 1$ in Algorithm 1. Let E be an elliptic curve defined by $\Omega \in \mathbb{H}_1$; a point $z \in E$ will be represented by the projective coordinates $(\theta_i(z))_{i\in Z(\overline{4})} = (\vartheta \left[\begin{smallmatrix} 0 \\ i/4 \end{smallmatrix}\right](z, \Omega/4))_{i\in Z(\overline{4})}$. Let $(\theta_i(0))_{i\in Z(\overline{4})} = (a, b, c, d)$. (In all the examples that we give, we assume that we are in a generic setting so that the formulas are well defined. Otherwise we can always choose a different subset of Riemann relations, at least in level 4).

3.2 Differential additions

Denote by $\widetilde{\mathscr{A}}$ the pullback of $\mathscr{A}$ via the natural projection $\kappa : \mathbb{A}^{n^g} \to \mathbb{P}^{n^g-1}$. In the following, we adopt the following convention: if $P = (P_i)_{i\in Z(\overline{n})}$ is a point of $\mathscr{A}$, we denote by $\widetilde{P} = (\widetilde{P}_i)_{i\in Z(\overline{n})}$ an affine lift of P that is a point $\widetilde{P}$ of $\mathbb{A}^{n^g}$ such that $\kappa(\widetilde{P}) = P$. We introduce the tilde notation in the formulas where we compute with affine points: it means that we want to distinguish two points lying on the same line cutting the origin of $\mathbb{A}^{n^g}$. We also choose once and for all an affine lift of the theta null point $\widetilde{0} = (\theta_i(0))_{i\in Z(\overline{n})}$ defined over K.

Next, for all n such that $2|n$, from the knowledge of $\widetilde{P} = (\widetilde{P}_i)_{i\in Z(\overline{n})}$, $\widetilde{Q} = (\widetilde{Q}_i)_{i\in Z(\overline{n})}$ and $\widetilde{(P - Q)} = (\widetilde{(P - Q)}_i)_{i\in Z(\overline{n})}$, the formula of Theorem 1, defines a unique $\widetilde{P + Q} = (\widetilde{(P + Q)}_i)_{i\in Z(\overline{n})}$ which is an affine lift of $P + Q$. Following [20], we write $\widetilde{P + Q} = \mathrm{DiffAdd}(\widetilde{P}, \widetilde{Q}, \widetilde{P - Q})$.

Algorithm 1: Normal addition algorithm in genus 1 level 4

input : The points $x = (x_0, x_1, x_2, x_3)$ and $y = (y_0, y_1, y_2, y_3)$ on E.
output : The projective coordinates $(\theta_i(x+y))$ of the point $t = x + y$.

return

$$t_0 = \frac{(x_0^2 + x_2^2)(y_0^2 + y_2^2)}{(a^2 + c^2)} + \frac{(x_0^2 - x_2^2)(y_0^2 - y_2^2)}{(a^2 - c^2)}$$
$$t_1 = \frac{(x_0x_1 + x_2x_3)(y_0y_1 + y_2y_3)}{(ab + cd)} + \frac{(x_0x_1 - x_2x_3)(y_0y_1 - y_2y_3)}{(ab - cd)}$$
$$t_2 = \frac{(x_1^2 + x_3^2)(y_1^2 + y_3^2)}{(a^2 + c^2)} + \frac{(x_1^2 - x_3^2)(y_1^2 - y_3^2)}{(a^2 - c^2)}$$
$$t_3 = \frac{(x_0x_3 + x_2x_1)(y_0y_3 + y_2y_1)}{(ac + bc)} + \frac{(x_0x_3 - x_2x_1)(y_0y_3 - y_2y_1)}{(ac - bc)}$$

Chaining the algorithm DiffAdd in a classical Montgomery ladder [4, Algorithm 9.5 p. 148] yields an algorithm that takes as inputs $\widetilde{Q} = (\widetilde{Q}_i)_{i \in Z(\overline{n})}$, $\widetilde{P+Q} = (\widetilde{(P+Q)}_i)_{i \in Z(\overline{n})}$, $\widetilde{P} = (\widetilde{P}_i)_{i \in Z(\overline{n})}$, $\widetilde{0} = (\widetilde{0}_i)_{i \in Z(\overline{n})}$ and an integer ℓ and outputs $\widetilde{Q + \ell P}$. We write $\widetilde{Q + \ell P} = \mathrm{ScalarMult}(\ell, \widetilde{P+Q}, \widetilde{P}, \widetilde{Q}, \widetilde{0})$. It is proved in [20] that the output of ScalarMult does not depend of the addition chain used to compute it.

There is a natural action by multiplication of $\overline{K}^*$ on the coordinate of a point in $\mathbb{A}^{n^g}$ that we denote by $\lambda * P$ for $\lambda \in \overline{K}^*$ and $P \in \mathbb{A}^{n^g}(\overline{K})$.

Lemma 1. *Let $P, Q \in \mathscr{A}(\overline{K})$ and let $\widetilde{P}, \widetilde{Q}, \widetilde{P+Q}$ be affine lifts of P, Q and $P+Q$. Let $\widetilde{R} = \mathrm{ScalarMult}(\ell, \widetilde{P+Q}, \widetilde{P}, \widetilde{Q}, \widetilde{0})$. Let $\alpha, \beta, \gamma, \delta \in \overline{K}^*$, we have [20, Remark 3]*

$$\mathrm{ScalarMult}(\ell, \alpha * \widetilde{P+Q}, \beta * \widetilde{P}, \gamma * \widetilde{Q}, \delta * \widetilde{0}) = (\alpha^\ell \beta^{\ell(\ell-1)} / \gamma^{\ell-1} \delta^{\ell(\ell-1)}) * \widetilde{R}, \quad (4)$$

$$\mathrm{ScalarMult}(\ell, \alpha * \widetilde{P}, \alpha * \widetilde{P}, \delta * \widetilde{0}, \delta * \widetilde{0}) = \frac{\alpha^{\ell^2}}{\delta^{\ell^2 - 1}} * \mathrm{ScalarMult}(\ell, \widetilde{P}, \widetilde{P}, \widetilde{0}, \widetilde{0}). \quad (5)$$

We illustrate differential additions with our previous $g = 1$, $n = 4$ example in Algorithm 2.

3.3 Normal additions in level 2

We consider the case $n = 2$ and we make the hypothesis that level 2 theta functions give a projectively normal embedding of the Kummer variety $\mathscr{K}_{\mathscr{A}}$ into $\mathbb{P}^{Z(\overline{2})}$. This condition is equivalent following [16, Corollary 4.5.2 and Remark (2)]

Algorithm 2: Differential addition algorithm in genus 1 level 4

input : The points $\widetilde{P} = (x_0, x_1, x_2, x_3)$, $\widetilde{Q} = (y_0, y_1, y_2, y_3)$ and $\widetilde{(P-Q)}(z_0, z_1, z_2, z_3)$ on $\widetilde{E}$.

output : The affines coordinates $\widetilde{P+Q} = (t_0, t_1, t_2, t_3)$.

return

$$
\begin{aligned}
t_0 &= \frac{(x_0^2 + x_2^2)(y_0^2 + y_2^2)}{z_0(a^2 + c^2)} + \frac{(x_0^2 - x_2^2)(y_0^2 - y_2^2)}{z_0(a^2 - c^2)} \\
t_1 &= \frac{(x_1^2 + x_3^2)(y_0^2 + y_2^2)}{z_1(a^2 + c^2)} + \frac{(x_1^2 - x_3^2)(y_0^2 - y_2^2)}{z_1(a^2 - c^2)} \\
t_2 &= \frac{(x_0^2 + x_2^2)(y_0^2 + y_2^2)}{z_2(a^2 + c^2)} - \frac{(x_0^2 - x_2^2)(y_0^2 - y_2^2)}{z_2(a^2 - c^2)} \\
t_3 &= \frac{(x_1^2 + x_3^2)(y_0^2 + y_2^2)}{z_3(a^2 + c^2)} - \frac{(x_1^2 - x_3^2)(y_0^2 - y_2^2)}{z_3(a^2 - c^2)}
\end{aligned}
$$

to the fact that for all $k, l \in Z(\overline{2})$ such that ${}^t k.l = 0$, $\theta_{k,l}(0) \neq 0$. As for $n = 2$, we can not distinguish a point Q from its opposite $-Q$, we can not expect to have a NormalAdd(P, Q) algorithm which returns a point since the result $P \pm Q$ is not determined from the input data. Nonetheless, there exists a NormalAdd$_2(P, Q)$ algorithm whose output uniquely determines the set of points $\{P + Q, P - Q\}$.

From the knowledge of $P = (\theta_i(z_P))_{i \in Z(\overline{2})}$ and $Q = (\theta_i(z_Q))_{i \in Z(\overline{2})}$, it is explained in [20, Section 5.2] that using Riemann's equations (3) with $z_3 = z_4 = 0$, we can recover for all $i, j \in Z(\overline{2})$

$$\kappa_{ij} = \theta_i(z_P + z_Q)\theta_j(z_P - z_Q) + \theta_j(z_P + z_Q)\theta_i(z_P - z_Q), \tag{6}$$

with the formula

$$
\begin{aligned}
&\kappa_{ij} = \\
&\sum_{\chi \in \hat{Z}(\overline{2})} \Bigg[\Big(\sum_{\eta \in Z(\overline{2})} \chi(\eta)\theta_{i+j+\eta}(0)\theta_{i-j+\eta}(0) \Big)^{-1} \Big(\sum_{\eta \in Z(\overline{2})} \chi(\eta)\theta_{i+k+\eta}(z_P)\theta_{i-k+\eta}(z_P) \Big) \\
&\qquad\qquad \Big(\sum_{\eta \in Z(\overline{2})} \chi(\eta)\theta_{j+l+\eta}(z_Q)\theta_{j-l+\eta}(z_Q) \Big) \Bigg].
\end{aligned} \tag{7}
$$

We can suppose that $\theta_0(z_P + z_Q)\theta_0(z_P - z_Q) \neq 0$, if necessary by replacing the index 0 by another one. For $i \in Z(\overline{2})$, if we let $\mathfrak{P}_i(X) = X^2 - 2\frac{\kappa_{i0}}{\kappa_{00}}X + \frac{\kappa_{ii}}{\kappa_{00}}$, then the roots of $\mathfrak{P}_i(X)$ are $\frac{\theta_i(z_P+z_Q)}{\theta_0(z_P+z_Q)}$, $\frac{\theta_i(z_P-z_Q)}{\theta_0(z_P-z_Q)} \in K$. If $P, Q \in \mathscr{K}_{\mathscr{A}}(\overline{K})$ are 2-torsion points, $P + Q = P - Q \in \mathscr{K}_{\mathscr{A}}(\overline{K})$ so each $\mathfrak{P}_i(X)$ has a double root. Otherwise, we can suppose that there exists $\alpha \in Z(\overline{2})$, $\alpha \neq 0$ such that the

matrix $M = \begin{pmatrix} \theta_0(z_P+z_Q) & \theta_0(z_P-z_Q) \\ \theta_\alpha(z_P+z_Q) & \theta_\alpha(z_P-z_Q) \end{pmatrix}$ is invertible. Consider the algebra $\mathfrak{A} = K[X]/(\mathfrak{P}_\alpha(X)) = K[x]$ and where x is the image of X is $\mathfrak{A}$ via the canonical projection. Then we can represent the set $\{\theta_\alpha(z_P+z_Q), \theta_\alpha(z_P-z_Q)\}$ by the set $\{x, 2\frac{\kappa_{\alpha 0}}{\kappa_{00}} - x\}$. Note that it implies the choice of a normalisation for $\theta_0(z_P+z_Q)$: in fact, in the case that $x = \theta_\alpha(z_P+z_Q)$ we have chosen $\theta_0(z_P+z_Q) = 1$ and in the case that $x = \theta_\alpha(z_P-z_Q)$, we have chosen $\theta_0(z_P-z_Q) = 1$. These choices are made possible by the hypothesis $\theta_0(z_P+z_Q)\theta_0(z_P-z_Q) \neq 0$.

We can then compute the couple $\{\gamma_i x + \delta_i, 2\frac{\kappa_{i0}}{\kappa_{00}} - \gamma_i x - \delta_i\}$ in $\mathfrak{A}$ representing the set $\{\theta_i(z_P+z_Q), \theta_i(z_P-z_Q)\}$ by solving the linear system with coefficients in $\mathfrak{A}$:

$$\begin{pmatrix} 1 & \frac{\kappa_{00}}{\kappa_{\alpha\alpha}} \\ x & \frac{\kappa_{\alpha 0}}{\kappa_{00}} - x \end{pmatrix} \begin{pmatrix} \theta_i(z_P-z_Q) \\ \theta_i(z_P+z_Q) \end{pmatrix} = \begin{pmatrix} \kappa_{i0} \\ \kappa_{i\alpha} \end{pmatrix}. \tag{8}$$

The algorithm NormalAdd$_2$ is then defined as follows: it takes as input $\widetilde{0} = (\theta_i(0))_{i\in Z(\overline{2})}$, $P = (\theta_i(z_P))_{i\in Z(\overline{2})}$ and $Q = (\theta_i(z_Q))_{i\in Z(\overline{2})}$ and outputs the polynomial defining the algebra $\mathfrak{A}$ and the $\gamma_i x + \delta_i$ for $i \in Z(\overline{2})$; defining the set $\{(\theta_i(z_P+z_Q))_{i\in Z(\overline{2})}, (\theta_i(z_P-z_Q))_{i\in Z(\overline{2})}\}$. It is clear that the output of NormalAdd$_2$ determines the set $\{P+Q, P-Q\}$ and that this algorithm only requires to compute a fixed number of operations in the base field.

We illustrate (the generic version of) this algorithm with $g = 1$ and $n = 2$ in Algorithm 3. Let E be an elliptic curve defined by $\Omega \in \mathbb{H}_1$; a point z of the Kummer line associated to E is represented by the projective coordinates $(\theta_i(z))_{i\in Z(\overline{2})} = (\vartheta\left[\begin{smallmatrix} 0 \\ i/2 \end{smallmatrix}\right](z, \Omega/2))_{i\in Z(\overline{2})}$. We let $(a,b) = (\theta_0(0), \theta_1(0)) = (\vartheta\left[\begin{smallmatrix} 0 \\ 0 \end{smallmatrix}\right](0, \Omega/2), \vartheta\left[\begin{smallmatrix} 0 \\ 1/2 \end{smallmatrix}\right](0, \Omega/2)$. We define $A = 2(a^2+b^2)$ and $B = 2(a^2-b^2)$. If $(a', b') = (\vartheta\left[\begin{smallmatrix} 0 \\ 0 \end{smallmatrix}\right](0, \Omega), \vartheta\left[\begin{smallmatrix} 1/2 \\ 0 \end{smallmatrix}\right](0, \Omega))$ is the coordinate of the dual theta null point, by the duplication formula we have that $A = 4(a')^2$ and $B = 4(b')^2$.

Algorithm 3: Normal addition algorithm in genus 1 level 2

input : The points $P = (x_0, x_1)$, $Q = (y_0, y_1)$ on E
output: The set $\{P+Q, P-Q\}$.

1 $\kappa_{00} = (x_0^2+x_1^2)(y_0^2+y_1^2)/A + (x_0^2-x_1^2)(y_0^2-y_1^2)/B$;
2 $\kappa_{11} = (x_0^2+x_1^2)(y_0^2+y_1^2)/A - (x_0^2-x_1^2)(y_0^2-y_1^2)/B$;
3 $\kappa_{01} = x_0x_1y_0y_1/ab$;
4 **return** $\{(\kappa_{00}, X), (\kappa_{00}, \kappa_{10} - X)\}$ where X is a root of $X^2 - \kappa_{10}X + \kappa_{00}\kappa_{11}$.

3.4 Differential additions in level 2

Differential additions are easier to handle: the algorithm outlined in Section 3.2 also works in level 2 with general points.

We illustrate this algorithm with our previous $g = 1$ and $n = 2$ example in Algorithm 4. Since this is the addition formula that will be used the most in the pairings algorithm, we give a factored form which may be more convenient for the computations. Note that when using a differential addition to compute ScalarMult, we can always choose affine lifts so that $a = x_0 = y_0 = 1$ in the notations of Algorithm 4. For the pairing algorithms that we will use, it is easy to see that we can also always multiply the formulas by the same (rational) value. This mean that we can replace the values (A, B) by $(1, B/A)$.

Algorithm 4: Differential Addition Algorithm in genus 1 level 2

input : The points $\widetilde{P} = (x_0, x_1)$, $\widetilde{Q} = (y_0, y_1)$ on $\widetilde{E}$, and $\widetilde{P - Q} = (z_0, z_1)$ with $z_0 z_1 \neq 0$.
output : The point $\widetilde{P + Q} = (t_0, t_1)$.

$t'_0 = (x_0^2 + x_1^2)(y_0^2 + y_2^2)/A$;
$t'_1 = (x_0^2 - x_1^2)(y_0^2 - y_1^2)/B$;
$t_0 = (t'_0 + t'_1)/z_0$;
$t_1 = (t'_0 - t'_1)/z_1$;
return (t_0, t_1)

In genus 2, the corresponding formulas are described in [10]. To handle the non generic case, one can use NormalAdd_2 to compute $\{P + Q, P - Q\}$; since we know $\widetilde{P - Q}$, it is easy to recover $\widetilde{P + Q}$.

3.5 Compatible additions in level 2

Using the NormalAdd_2 algorithm, it is also possible to recover $P + R$ from the knowledge of $P, Q, R, P + Q, Q + R$ on a Kummer variety when Q is not a point of two torsion. We call this operation compatible addition and we note $P + R = \text{CompatAdd}(P, Q, R, P + Q, Q + R)$. The idea is the following: $\text{NormalAdd}_2(P, R)$ gives the set $\{P + R, P - R\}$ and we want to be able to identify $P + R$ in this set. We can suppose that $2P \neq 0$ and $2R \neq 0$ because if this is not the case we have $P + R = P - R$ in the Kummer variety. We remark that $\text{NormalAdd}_2(P + R, P + Q)$ determines $\{2P + Q + R, Q - R\}$ and $\text{NormalAdd}_2(P - R, P + Q)$ gives the set $\{2P + Q - R, Q + R\}$. As we know $Q + R$ and as $2P \neq 0$ and $2Q \neq 0$ by hypothesis, we have a way to distinguish $P + R$ from $P - R$.

We now describe an algorithm to compute the compatible addition. Let $\mathfrak{P}_\alpha$ be the defining polynomial of the algebra $\mathfrak{A} = K[X]/(\mathfrak{P}_\alpha) = K[x]$ given by $\text{NormalAdd}_2(P, R)$. We want to find a root of $\mathfrak{P}_\alpha$ from the knowledge of $P + Q$ and $Q + R$ without computing a square root in K. We use for all $i \in Z(\overline{2})$, $\theta_i(z_{P \pm R}) = \gamma_i x + \delta_i$ as a convenient notation. Then we define $\kappa(x)_{ij}$ as the elements of the algebra $\mathfrak{A}$ given by equation (7) where we have replaced z_P by z_{P+Q} and z_Q by $z_{P \pm R}$. We can suppose that $\kappa(x)_{00} \neq 0$ if necessary by changing

this index and we consider the polynomials $\mathfrak{Q}_i(Y) = Y^2 - 2\frac{\kappa(x)_{i0}}{\kappa(x)_{00}}Y + \frac{\kappa(x)_{ii}}{\kappa(x)_{00}}$ for $i \in Z(\overline{2})$. By representing an elements of $\mathfrak{A}$ as a couple a elements of K via the natural evaluation morphism at the roots of $\mathfrak{P}_\alpha(X)$, the roots of the $\mathfrak{Q}_i(Y)$ are given by the couples $\{\frac{\theta_i(z_{2P+Q+R})}{\theta_0(z_{2P+Q+R})}, \frac{\theta_i(z_{Q-R})}{\theta_0(z_{Q-R})}\}$,$\{\frac{\theta_i(z_{2P+Q-R})}{\theta_0(z_{2P+Q+R})}, \frac{\theta_i(z_{Q+R})}{\theta_0(z_{Q+R})}\}$. As consequence, if we evaluate $\mathfrak{Q}_i(Y)$ at $\frac{\theta_i(z_{Q+R})}{\theta_0(z_{Q+R})}$, we obtain a non invertible element of $\mathfrak{A}$ which is non nul since $2P \neq 0$. As $\frac{\theta_i(z_{Q+R})}{\theta_0(z_{Q+R})}$ is a coordinate of $P+Q$, we are done. In the following, we use the notation $P+R = \mathrm{CompatAdd}(P, Q, R, P+Q, Q+R)$ for the preceding algorithm. It is clear from our discussion that this algorithm only requires a fixed number of operations in the base field.

We illustrate Compatible additions once more with our $g = 1$ and $n = 2$ example in Algorithm 5.

Algorithm 5: Compatible additions in genus 1 level 2

input : x, y, $Y = x + z$, $X = y + z$.
output : $Z = x + y$.

Computing $x \pm y$;
$\alpha = (x_0^2 + x_1^2)(y_0^2 + y_1^2)/A$;
$\beta = (x_0^2 - x_1^2)(y_0^2 - y_1^2)/B$;
$\kappa_{00} = (\alpha + \beta), \kappa_{11} = (\alpha - \beta)$;
$\kappa_{10} := x_0 x_1 y_0 y_1 / ab$;
Computing $(x + z) \pm (y + z)$;
$\alpha' = (Y_0^2 + Y_1^2)(X_0^2 + X_1^2)/A$;
$\beta' = (Y_0^2 - Y_1^2)(X_0^2 - X_1^2)/B$;
$\kappa'_{00} = \alpha' + \beta', \kappa'_{11} = \alpha' - \beta'$;
$\kappa'_{10} = Y_1 Y_2 X_1 X_2 / ab$;
return
$x + y = [\kappa_{00}(\kappa_{10}\kappa'_{00} - \kappa'_{10}\kappa_{00}), \kappa_{10}(\kappa_{10}\kappa'_{00} - \kappa'_{10}\kappa_{00}) + \kappa_{00}(\kappa_{11}\kappa'_{00} - \kappa'_{11}\kappa_{00})]$

3.6 Three way additions (in level 2 or more)

The three way addition ThreeAdd is an important ingredient to compute the Miller functions $\mathfrak{f}_{\lambda,\mu,n,P}$ of Lemma 4 below. From P, Q, R, $P+Q$, $P+R$, $Q+R$ on a Kummer variety given in the coordinate system provided by level 2 theta functions, it allows to compute $P+Q+R$. It is also useful when the level n is greater than two: given any affine lifts $\widetilde{P}$, $\widetilde{Q}$, $\widetilde{R}$, $\widetilde{P+Q}$, $\widetilde{P+R}$, $\widetilde{Q+R}$ in $\widetilde{\mathscr{A}}$, it allows to compute a "compatible lift" $\widetilde{P+Q+R} \in \widetilde{\mathscr{A}}$.

In the following, we say that a property is true for $R \in \mathscr{A}(\mathbb{C})$ (resp. $z \in \mathbb{C}^g$) a general point of $\mathscr{A}$ (resp. of $\mathbb{C}^g$) if it is verified for all R (resp. z) taken in a Zariski dense subset of $\mathscr{A}$ (resp. in $\pi^{-1}(U)$ where U is a Zariski dense subset of $\mathscr{A}$). For instance, if $f \in K(\mathscr{A})$, we say that we can evaluate f at a general point

of $\mathscr{A}$, if we have an algorithm to compute $f(R)$ for all $R \in U$, U a Zariski dense subset of $\mathscr{A}$. Then we can state the proposition:

Proposition 1. *Suppose that $2 \mid n$ and let $z_1, z_2 \in \mathbb{C}^g$. For a general point $z \in \mathbb{C}^g$, suppose that we are given the (affine) theta coordinates of level n of z_1, z_2, z, z_1+z_2, z_1+z, z_2+z. Then one can recover the affine coordinates $(\theta_i(z_1+z_2+z)_{i\in Z(\overline{n})})$ of z_1+z_2+z. Furthermore, if $4 \mid n$, then the Proposition holds for any $z \in \mathbb{C}^g$.*

Proof. For $i,j,k,l \in Z(\overline{2n})$ such that $i+j$, $i+k$ and $i+l \in Z(\overline{n})$, $\chi \in \hat{Z}(\overline{2})$, let:

$$\begin{aligned}
L_1(z,i,j,\chi) &= \sum_{\eta\in Z(\overline{2})} \chi(\eta)\theta_{i+j+\eta}(z+z_1+z_2)\theta_{i-j+\eta}(z_1),\\
L_2(z,k,l,\chi) &= \sum_{\eta\in Z(\overline{2})} \chi(\eta)\theta_{k+l+\eta}(z_2)\theta_{k-l+\eta}(z),\\
L_3(z,i,k,\chi) &= \sum_{\eta\in Z(\overline{2})} \chi(\eta)\theta_{i+k+\eta}(0)\theta_{i-k+\eta}(z_2+z),\\
L_4(z,j,l,\chi) &= \sum_{\eta\in Z(\overline{2})} \chi(\eta)\theta_{j+l+\eta}(z_1+z)\theta_{j-l+\eta}(z_1+z_2)
\end{aligned}$$

Then, by Theorem 1, we have

$$L_1(z,i,j,\chi).L_2(z,k,l,\chi) = L_3(z,i,k,\chi).L_4(z,j,l,\chi). \tag{9}$$

As $2 \mid n$, we know that the linear system associated to Θ_n has no base point. Thus, if we fix i,j with $i+j \in Z(\overline{n})$, we can choose k,l such that $i+k$ and $i+l \in Z(\overline{n})$ and $\theta_{k+l}(z_2) \neq 0$. Suppose that $L_2(z,k,l,\chi) = 0$ for z a general point of $\mathbb{C}^g$. Then, $L_2(z,k,l,\chi)$ is a non trivial degree 1 relation between theta functions. But such a relation does not exist since it is known that $(\theta_i(z))_{i\in Z(\overline{n})}$ form a basis of the vector space of L_Ω-quasi-periodic functions of weight n [25, Proposition 1.3 p. 122]. So we can assume (for a general point z of $\mathbb{C}^g$) that for all χ, $L_2(z,k,l,\chi) \neq 0$.

We can then compute $L_1(z,i,j,\chi)$ for a general $z \in \mathbb{C}^g$ and for all $\chi \in \hat{Z}(\overline{2})$. By summing over the characters, we can thus compute for all $i,j \in Z(\overline{2n})$ such that $i+j \in Z(\overline{n})$ the products $\theta_{i+j}(z+z_1+z_2)\theta_{i-j}(z_1)$. Using again the fact that the linear system associated to Θ_n has no base point, we obtain that we can compute $(\theta_i(z+z_1+z_2))_{i\in Z(\overline{n})}$ for z a general point of $\mathbb{C}^g$.

If $4 \mid n$, to show that we can compute $L_1(z,i,j,\chi)$ for any $z \in \mathbb{C}^g$, we have to show that $L_2(z,k,l,\chi) \neq 0$ for some $k,l \in Z(\overline{n})$ such that $i+j+k+l \in 2Z(\overline{n})$. By the duplication formula [15, Theorem 2, p. 139–141], with a slight abuse of notations, we get that $L_2(z,k,l,\chi) = 2^g\theta\left[\begin{smallmatrix}\frac{\chi}{2}\\ \frac{k+l}{2n}\end{smallmatrix}\right](z_2+z, 2\frac{\Omega}{n})\theta\left[\begin{smallmatrix}\frac{\chi}{2}\\ \frac{k-l}{2n}\end{smallmatrix}\right](z_2-z, 2\frac{\Omega}{n})$. But by [22, Result a) p. 340], for any $a,b \in \mathbb{Z}$ and $z \in \mathbb{C}^g$, there exists $c \in \mathbb{Z}$ such that $\theta\left[\begin{smallmatrix}\frac{a}{2}\\ \frac{b}{2n}+\frac{c}{4}\end{smallmatrix}\right](z, 2\frac{\Omega}{n}) \neq 0$. As for any $k,l \in Z(\overline{n})$, we can find $k',l' \in Z(\overline{4})$ such that $L_2(z,k+k'+l',l+k'-l',\chi) \neq 0$ and we are done.

Definition 1. *Let $\widetilde{P}$, $\widetilde{Q}$, $\widetilde{R}$, $\widetilde{P+Q}$, $\widetilde{P+R}$, $\widetilde{Q+R}$ in $\widetilde{\mathscr{A}}$ be given by their theta functions of level n. Let $\widetilde{P+Q+R} \in \widetilde{\mathscr{A}}$ be the point computed using the relations of the proof of Proposition 1.*

We note $\widetilde{P+Q+R} = \mathrm{ThreeAdd}(\widetilde{P}, \widetilde{Q}, \widetilde{R}, \widetilde{Q+R}, \widetilde{P+R}, \widetilde{P+Q})$.

From the way the three addition is computed, we immediately get the multiplicative action of Section 3.2.

Lemma 2. *Let* $\widetilde{T} = \mathrm{ThreeAdd}(\widetilde{P}, \widetilde{Q}, \widetilde{R}, \widetilde{Q+R}, \widetilde{P+R}, \widetilde{P+Q})$. *Then*

$$\mathrm{ThreeAdd}(\alpha * \widetilde{P}, \beta * \widetilde{Q}, \gamma * \widetilde{R}, A * \widetilde{Q+R}, B * \widetilde{P+R}, C * \widetilde{P+Q}) = \frac{ABC}{\alpha\beta\gamma} * \widetilde{T}.$$

We illustrate Three way additions with our $g = 1$, $n = 2$ example in Algorithm 6.

Algorithm 6: Three way addition in genus 1 level 2

input : The points $x, y, z, X = y + z, Y = x + z, Z = x + y$ on E.
output : $T = x + y + z$.

return

$$T_0 = \frac{(aX_0 + bX_1)(Y_0Z_0 + Y_1Z_1)}{x_0(y_0z_0 + y_1z_1)} + \frac{(aX_0 - bX_1)(Y_0Z_0 - Y_1Z_1)}{x_0(y_0z_0 - y_1z_1)}$$
$$T_1 = \frac{(aX_0 + bX_1)(Y_0Z_0 + Y_1Z_1)}{x_1(y_0z_0 + y_1z_1)} - \frac{(aX_0 - bX_1)(Y_0Z_0 - Y_1Z_1)}{x_1(y_0z_0 - y_1z_1)}$$

4 A generalisation of Miller's algorithm

In this section, we first recall the classical Miller's algorithm to compute the function of an elliptic curve defined up to a constant factor by a principal divisor, then we explain how to generalize this algorithm to all abelian varieties.

Let $\mathscr{E}$ be an elliptic curve and denote by 0 the zero element of the group law of $\mathscr{E}$. Let P be a point of $\mathscr{E}$ of order ℓ. Then by [27, Corollary 3.5 p. 67] the divisor $\ell(P) - \ell(0)$ is principal and we denote by f an element of $K(\mathscr{E})$ defined up to a constant factor by $(f) = \ell(P) - \ell(0)$. More generally, let $f_{\lambda,P} \in K(\mathscr{E})$ be defined up to a constant factor by

$$(f_{\lambda,P}) = \lambda(P) - ([\lambda]P) + (\lambda - 1)(0).$$

Miller's algorithm [21] to compute f is based on the following remark. For $\lambda > 0$ an integer, we have:

$$f_{\lambda+\mu,P} = f_{\lambda,P} f_{\mu,P} \mathfrak{f}_{\lambda,\mu,P},$$

where $\mathfrak{f}_{\lambda,\mu,P}$ is a function associated to the divisor

$$([\lambda+\mu]P)-([\lambda]P)-([\mu]P)+(0),$$

assuming a suitable normalisation of the functions. If $\mathscr{E}$ is given by a Weierstrass equation, $\mathfrak{f}_{\lambda,\mu,P}$ can be computed easily from the usual cord and tangent definition of the group law on $\mathscr{E}$. As $f_{1,P}$ is a constant function, we obtain an efficient square and multiply algorithm to compute $f_{\ell,P}$.

Now, let $\mathscr{A}$ be a principally polarised abelian variety defined by a period matrix Ω. We recall that we denote by Θ the theta divisor of $\mathscr{A}$ defined by Ω which is the zero divisor of the associated Riemann theta function and by Θ_n the theta divisor of level n (as in Section 2). Denote by 0 the zero element of $\mathscr{A}$ and let P be a point of order ℓ. For $\lambda>0$ an integer, by an easy induction using the theorem of the square [24, Corollary 4, p. 67], we see that the divisor

$$\lambda\tau_P^*\Theta_n-\tau_{\lambda P}^*\Theta_n+(\lambda-1)\Theta_n \tag{10}$$

is principal. We denote by $f_{\lambda,n,P}$ or more simply by $f_{\lambda,P}$ if $n=1$ the function defined up to a constant factor by this divisor. In the following, we will use the same notation to denote $f_{\lambda,n,P}$ or its pullback by the projection $\pi:\mathbb{C}^g\to\mathscr{A}$.

As noted, $f_{\lambda,n,P}$ is defined only up to a constant factor. This will not present a problem since we will only evaluate this function on degree zero cycles. In practice, one usually take a unique representative in the class of functions defined up to a constant factor by imposing a normalizing condition at a point (for instance at 0). Also the definition of $f_{\lambda,n,P}$ depends on the divisor Θ_n. The divisor of any theta function of level n is linearly equivalent to Θ_n, but the corresponding function $f_{\lambda,n,P}$ is be different. However, we will see in Section 5 that the definition of the Weil and Tate pairings depends only on the equivalence class of Θ_n.

Lemma 3. *Let $z_P\in\mathbb{C}^g$ be such that $P=\pi(z_P)$. For all $i\in Z(\overline{n})$, we have up to a constant factor:*

$$f_{\lambda,n,P}(z)=\frac{\theta_i(z)}{\theta_i(z+\lambda z_P)}\left(\frac{\theta_i(z+z_P)}{\theta_i(z)}\right)^{\lambda}, \tag{11}$$

where for $i\in Z(\overline{n})$, $\theta_i(z)=\theta\left[\begin{smallmatrix}0\\ i/n\end{smallmatrix}\right](z,\Omega/n)$ are the theta functions of level n.

Proof. Denote by $g_{\lambda,n,P}$ the function with domain $\mathbb{C}^g$ given by the right hand of (11). First, using the quasi-periodicity properties of theta functions with characteristics (2), we obtain that for all $\mu,\nu\in\mathbb{Z}^g$ and for all $z\in\mathbb{C}^g$ we have

$$g_{\lambda,n,P}(z+\mu+\nu\Omega)=g_{\lambda,n,P}(z).$$

As a consequence, $g_{\lambda,n,P}$ descends to a well defined function of $\mathscr{A}$. As for all $i\in Z(\overline{n})$, the zero divisor of $\theta_i(z)$ is linearly equivalent to Θ_n, it is clear that the zero divisor of $g_{\lambda,n,P}$ is (10) (up to replacing Θ_n by an equivalent divisor).

Keeping the notations of 3, we deduce by an immediate computation using formula (11) that:

Lemma 4. *For all λ, μ positive integers, we have up to a constant factor*

$$f_{\lambda.\mu,n,P} = f^{\lambda}_{\mu,n,P}.f_{\lambda,n,\mu.P}. \tag{12}$$

We also have the following relation

$$f_{\lambda+\mu,n,P} = f_{\lambda,n,P} f_{\mu,n,P} \mathfrak{f}_{\lambda,\mu,n,P}, \tag{13}$$

where $\mathfrak{f}_{\lambda,\mu,n,P}$ is a function associated to the divisor $\tau^*_{(\lambda+\mu)P}\Theta_n - \tau^*_{\lambda P}\Theta_n - \tau^*_{\mu P}\Theta_n + \Theta_n$. The function $\mathfrak{f}_{\lambda,\mu,n,P}$ is uniquely defined if we impose that it is normalized on a point. From the definition, we have by using Lemma 3

Lemma 5. *Let $z_P \in \mathbb{C}^g$ be such that $P = \pi(z_P)$. Let $i \in Z(\overline{n})$. For all λ, μ positive integers, we have up to a constant factor*

$$\mathfrak{f}_{\lambda,\mu,n,P}(z) = \frac{\theta_i(z + \lambda z_P)\theta_i(z + \mu z_P)}{\theta_i(z + (\lambda+\mu)z_P)\theta_i(z)} \tag{14}$$

(for all z where $\mathfrak{f}_{\lambda,\mu,n,P}$ is defined).

We can now explain how the various addition algorithms presented in Section 3 allow us to compute a normalized version of the function $\mathfrak{f}_{\lambda,\mu,n,P}(z)$ on a general point.

Proposition 2. *Suppose that $4 \mid n$ and let $\lambda, \mu \in \mathbb{N}$. Suppose that we are given $\theta_i(\lambda z_P)$, $\theta_i(\mu z_P)$ and $\theta_i(0)$ for $i \in Z(\overline{n})$. Suppose that we can evaluate the functions $\theta_i(z)$, $\theta_i(z + \lambda z_P)$ and $\theta_i(z + \mu z_P)$ for all $i \in Z(\overline{n})$ at a point $z \in \mathbb{C}^g$. Then we can evaluate the projective coordinates $(\theta_i(z + (\lambda+\mu)z_P))_{i \in Z(\overline{n})}$.*

If $2 \mid n$, the Proposition also holds for a general point $z \in \mathbb{C}^g$.

Proof. If we had the affine coordinates $(\theta_i((\lambda+\mu)z_P))_{i \in Z(\overline{n})}$ then by Proposition 1 one could recover the affine coordinates $(\theta_i(z + (\lambda+\mu)z_P))_{i \in Z(\overline{n})}$ using the three way additions. But by Lemma 2, if we can only compute the projective coordinates $(\theta_i((\lambda+\mu)z_P))_{i \in Z(\overline{n})}$, then the three way addition gives us the projective coordinates of $(\theta_i(z + (\lambda+\mu)z_P))_{i \in Z(\overline{n})}$.

If $4|n$, one can then use NormalAdd to compute $(\theta_i(\lambda+\mu)z_P)_{i \in Z(\overline{n})}$ from $(\theta_i(\lambda z_P))_{i \in Z(\overline{n})}$ and $(\theta_i(\mu z_P))_{i \in Z(\overline{n})}$. In the case that $n = 2$, one need to use the CompatAdd algorithm instead to recover $(\theta_i(\lambda+\mu)z_P)_{i \in Z(\overline{n})}$ from the knowledge of $(\theta_i(\lambda z_P))_{i \in Z(\overline{n})}$, $(\theta_i(\mu z_P))_{i \in Z(\overline{n})}$, $(\theta_i(z))_{i \in Z(\overline{n})}$, $(\theta_i(z + \lambda z_P))_{i \in Z(\overline{n})}$, $(\theta_i(z + \mu z_P))_{i \in Z(\overline{n})}$ for z a general point of $\mathbb{C}^g$.

When the level n is divisible by 4, we can use this proposition to evaluate a normalized function $\mathfrak{f}_{\lambda,\mu,n,P}$.

Corollary 1. *Let $4 \mid n$, $Q \in \mathscr{A}$, and let $R \in \mathscr{A}$ be a point such that neither R or $Q + R$ is a pole or zero of the divisor of $\mathfrak{f}_{\lambda,\mu,n,P}$. Then from the knowledge of λP and μP, we can evaluate $\mathfrak{f}_{\lambda,\mu,n,P}(Q+R)/\mathfrak{f}_{\lambda,\mu,n,P}(R)$.*

Proof. We fix affine lifts $\widetilde{\lambda P}$ and $\widetilde{\mu P}$ of λP and μP. We compute $(\lambda+\mu)P$ using NormalAdd algorithm and chose an affine lift $\widetilde{(\lambda+\mu)P}$. For a point $X = Q+R, R$, we compute $X+\lambda P$, $X+\mu P$ using NormalAdd and choose any affine lifts $\widetilde{X}$, $\widetilde{X+\lambda P}$ and $\widetilde{X+\mu P}$. Using the three way add, we get an affine lift $X+\widetilde{\lambda P}+\mu P$ of $X+\lambda P+\mu P$.

For $T \in \{\lambda P, \mu P, X, (\lambda+\mu)P, \lambda P+X, \mu P+X, X+(\lambda+\mu)P\}$, let $z_T \in \mathbb{C}^g$ be such that $T = \pi(z_T)$ and let $\alpha_T \in \overline{K}$ be such that for all $i \in Z(\overline{n})$, $\theta_i(z_T) = \alpha_T * T_i$ (where $*$ is the multiplicative action described in Section 3.1).

By Lemma 2, we have

$$\alpha_{X+(\lambda+\mu)P} = \frac{\alpha_{X+\lambda P}\alpha_{X+\mu P}}{\alpha_X}\frac{\alpha_{(\lambda+\mu)P}}{\alpha_{\lambda P}\alpha_{\mu P}}.$$

In particular, for $i \in Z(\overline{n})$, the quotient $\frac{(\widetilde{X+\lambda P})_i(\widetilde{X+\mu P})_i}{(X+\widetilde{(\lambda+\mu)P})_i(\widetilde{X})_i}$ (if defined) does not depend on choice of an affine lift for X, $X+\lambda P$ and $X+\mu P$, but only on the choises of $\widetilde{\lambda P}$, $\widetilde{\mu P}$ and $\widetilde{(\lambda+\mu)P}$. By applying that with $X = Q+R$ et $X = R$ we obtain that the function $\mathfrak{f}_{\lambda,\mu,n,P}$ evaluated at the cycle $(Q+R)-(R)$ is given by

$$\frac{(Q+\widetilde{R+\lambda P})_i(Q+\widetilde{R+\mu P})_i}{(Q+R+\widetilde{(\lambda+\mu)P})_i(\widetilde{Q+R})_i}\frac{(R+\widetilde{(\lambda+\mu)P})_i(\widetilde{R})_i}{(\widetilde{R+\lambda P})_i(\widetilde{R+\mu P})_i},$$

for $i \in Z(\overline{n})$ such that this fraction is defined.

If we go back to the definition of $f_{\lambda,n,P}$ given by Lemma 3, the method of [20] provides us with another way to compute it without going through the $\mathfrak{f}_{\lambda,\mu,n,P}$ functions. We assume in the Proposition that $f_{\lambda,n,P}$ is well defined on the cycle $(Q)-(0)$ (as we will see in Section 5 this is usually the case), we leave to the reader the easy adaptation to make to evaluate it on the cycle $(Q+R)-(R)$.

Proposition 3. *Let $\widetilde{P}$, $\widetilde{Q}$ and $\widetilde{0}$ be affine lifts of P,Q and 0. Let $\widetilde{P+Q}$ be a lift of $P+Q$ (if $4 \mid n$, we can compute it with a normal addition, otherwise we have to assume it is given). Note $\widetilde{Q+\lambda P} = \mathrm{ScalarMult}(\lambda, \widetilde{P+Q}, \widetilde{P}, \widetilde{Q}, \widetilde{0})$ and $\widetilde{\lambda P} = \mathrm{ScalarMult}(\lambda, \widetilde{P}, \widetilde{P}, \widetilde{0}, \widetilde{0})$. We have*

$$\frac{f_{\lambda,n,P}(z_Q)}{f_{\lambda,n,P}(0)} = \frac{\widetilde{Q}_i.\widetilde{\lambda P}_i}{(\widetilde{Q+\lambda P})_i.\widetilde{0}_i}\left(\frac{(\widetilde{P+Q})_i.\widetilde{0}_i}{\widetilde{Q}_i.\widetilde{P}_i}\right)^{\lambda} \tag{15}$$

Proof. Let $\alpha, \beta, \gamma, \delta \in \overline{K}^*$ be such that $\alpha * (\widetilde{P+Q}) = (\theta_i(z_P+z_Q))$, $\beta * \widetilde{P} = (\theta_i(z_P))$, $\gamma * \widetilde{Q} = (\theta_i(z_q))$ and $\delta * \widetilde{0} = (\theta_i(0))$.

By definition, we have

$$\frac{f_{\lambda,n,P}(z_Q)}{f_{\lambda,n,P}(0)} = \frac{\theta_i(z_Q)\theta_i(\lambda z_P)}{\theta_i(z_Q+\lambda z_P)\theta_i(0)}\left(\frac{\theta_i(z_Q+z_P)\theta_i(0)}{\theta_i(z_Q)\theta_i(z_P)}\right)^{\lambda} \tag{16}$$

Now using (4) and (5), we see that the right hand of (15) is equal to the right hand of (16) up to the factor

$$\frac{\gamma\beta^{\lambda^2}\delta^{\lambda(\lambda-1)}}{\frac{\alpha^\lambda\beta^{\lambda(\lambda-1)}}{\gamma^{\lambda-1}}\delta^{\lambda^2-1}\delta}\left(\frac{\alpha\delta}{\beta\gamma}\right)^\lambda = 1.$$

Since almost all known variations of pairing computation algorithms use the Miller's functions $f_{\lambda,n,P}$ and $\mathfrak{f}_{\lambda,\mu,n,P}$, we see that we can extend them to all abelian varieties with Corollary 1, at least if the level is divisible by 4. In the following, we make explicit how to transcribe the Weil, Tate, ate and optimal ate pairings, explain some optimizations and work out the case of level two.

5 The Weil and Tate pairings

In this section, we recall the definition of the Weil and Tate pairings in the general context of abelian varieties. There are several definitions of the Weil pairing leading to different formulas with their own interest in regard to algorithmic applications. Most of the proofs of the equivalence between these definitions rely on Weil's reciprocity theorem. We explain that a generalisation due to Lang of the Weil's reciprocity allows to adapt the usual proofs with minor modifications.

5.1 The Weil pairing

Let $f : \mathscr{A} \to \mathscr{B}$ be a separable isogeny with kernel L between two abelian varieties defined over k. Then we have the following diagram:

$$\begin{array}{ccccccccc} 0 & \longrightarrow & L & \longrightarrow & \mathscr{A} & \xrightarrow{f} & \mathscr{B} & \longrightarrow & 0 \\ & & 0 & \longleftarrow & \hat{\mathscr{A}} & \xleftarrow{\hat{f}} & \hat{\mathscr{B}} & \longleftarrow & \hat{L} & \longleftarrow & 0 \end{array}$$

The Kernel $\hat{L}$ is the Cartier dual of L, so that we have a non degenerate pairing $e_f : L \times \hat{L} \to \overline{K}^*$.

We can give an explicit description of this pairing. If $Q \in \hat{L}(\overline{K})$, Q defines a divisor D_Q on $\mathscr{B}$ modulo linear equivalence. Then $\hat{f}(Q) = 0$ corresponds to f^*D_Q so there is a function g_Q on $\mathscr{A}$ such that $f^*D_Q = \sum_{P\in L(\overline{K})} \tau_P^* D_Q = (g_Q)$. Then for all $P \in L(\overline{K})$, as (g_Q) is invariant by translation by P, $g_Q(x)/g_Q(x+P)$ is a constant function. Its definition does not depend on the choice of D_Q and g_Q and we have $e_f(P,Q) = g_Q(x)/g_Q(x+P)$.

Applying this to the isogeny $[\ell] : \mathscr{A} \to \mathscr{A}$, we recover the Weil pairing $e_W : \mathscr{A}[\ell] \times \hat{\mathscr{A}}[\ell] \to \mu_\ell$ where μ_ℓ is the set of ℓ^{th}-roots of unity in $\overline{K}$. We suppose that $\mathscr{A}$ has a principal polarisation Θ. Composing with the polarization $\varphi : \mathscr{A} \to \hat{\mathscr{A}}$ associated to the divisor Θ, we get the Weil pairing as $e_W : \mathscr{A}[\ell] \times \mathscr{A}[\ell] \to \mu_\ell$.

When $\mathscr{A}$ is an elliptic curve, it is well known that the Weil pairing can be computed as

$$e_W(P,Q) = \frac{f_{\ell,P}((Q)-(0))}{f_{\ell,Q}((P)-(0))}. \tag{17}$$

This result can be proved with Weil's reciprocity theorem. It can be generalized to the case where $\mathscr{A}$ is the Jacobian of a curve (see for instance [4]), which is the usual setting in cryptography, because the points and the group law on $\mathscr{A}$ have a convenient representation in term of divisors on the curve.

By using Lang's reciprocity theorem [17, Theorem 4], it is possible to obtain similar results in the general context of an abelian variety $\mathscr{A}$ with a principal polarisation $\varphi : \mathscr{A} \to \hat{\mathscr{A}}$ with minor adaptations of the proofs. To explain this, we denote by $S : Z^0(\mathscr{A}) \to \mathscr{A}(\overline{K})$, the morphism given by $\sum n_i(P_i) \mapsto \sum n_i P_i$. If $Z = \sum n_i(P_i) \in Z^0(\mathscr{A})$, we let $\varphi(Z) = \sum n_i(\varphi(P_i)) \in Z^0(\hat{\mathscr{A}})$. The cycle $\varphi(Z)$ defines a line bundle on $\mathscr{A}$ associated to the divisor $D_Z = \sum n_i(\tau_{P_i}^*\Theta - \Theta)$. If $S(Z) = 0$, this line bundle is linearly equivalent to 0 by the theorem of the square. This means that D_Z is the divisor of a function, defined up to a constant factor, that we denote by f_Z (the constant factor will play no role in the following since we only consider evaluations of f_Z on degree zero cycles). With these notations, we have

Proposition 4 (Lang reciprocity). *Let $Z_1, Z_2 \in Z^0(\mathscr{A})$ be such that $S(Z_i) = 0$ for $i = 1, 2$. Suppose that $\mathrm{Supp}(Z_1) \cap (f_{Z_2}) = \mathrm{Supp}(Z_2) \cap (f_{Z_1}) = \varnothing$, then we have $f_{Z_1}(Z_2) = f_{Z_2}(Z_1)$.*

Proof. Let $Z_1' = \varphi(Z_1) \in Z^0(\hat{\mathscr{A}})$. Via the canonical isomorphism $\mathscr{A} \to \hat{\hat{\mathscr{A}}}$, $Z_2 \in Z^0(\mathscr{A})$ defines as above a divisor on $\hat{\mathscr{A}}$ and because $S(Z_2) = 0$ this divisor D'_{Z_2} is linearly equivalent to 0. Denote by $f'_{Z_2} \in K(\hat{\mathscr{A}})$ a function defined up to a constant factor by D'_{Z_2}. Let $D_{\mathcal{P}}$ be a divisor associated to a Pointcarré line bundle of $\hat{\mathscr{A}} \times \mathscr{A}$. Then by [17, Theorem 4] applied to the divisorial correspondance given by $D_{\mathcal{P}}$, we have $f_{Z_1}(Z_2) = f'_{Z_2}(Z_1')$. In fact it is clear that f_{Z_1} is nothing but $D_{\mathcal{P}}(Z_1')$ with the notations of [17] and because a Pointcarré bundle parametrising line bundles of $\mathscr{A}$ via the first projection over $\hat{\mathscr{A}}$ is a Pointcarré bundle parametrising line bundles of $\hat{\mathscr{A}}$ via the second projection over $\mathscr{A} = \hat{\hat{\mathscr{A}}}$, we have $f'_{Z_2} =^t D_{\mathcal{P}}(Z_2)$.

To finish the proof, it remains to show that $f'_{Z_2}(Z_1') = f_{Z_2}(Z_1)$ but this is an immediate consequence of the fact that $\varphi^*(f'_{Z_2}) = f_{Z_2}$.

In order to show the usefullness of the proposition, we prove the following theorem by adapting the proof of the same result in the case of elliptic curves given in [7].

Theorem 2. *Let $P, Q \in \mathscr{A}[\ell]$. Let D_P and D_Q be two cycles equivalent to $(P)-(0)$ and $(Q)-(0)$. The Weil pairing is given by*

$$e_W(P,Q) = \frac{f_{\ell D_P}(D_Q)}{f_{\ell D_Q}(D_P)}. \tag{18}$$

Proof. Let P_0 (resp. Q_0) be a point such that $P = \ell P_0$ (resp. $\ell Q_0 = Q$). For $X = P, Q$, we have $[\ell]^* D_X = \sum_{R \in \mathscr{A}[\ell]} (X_0 + R) - (R)$ and $S(\ell D_X) = S([\ell]^* D_X) = 0$. Thus, we can set $g_X = f_{[\ell]^* D_X}$. For $X = P, Q$, it is clear by comparing the divisors that, up to a constant factor, we have $g_X^\ell = [\ell]^* f_{D_X}$.

Let $Z_P = (\ell - 1)(P_0) + (P_0 - P) - \ell(0) \in Z^0(\mathscr{A})$ and let $h_P = f_{Z_P}$ as $S(Z_P) = 0$. Let $H_P = \prod_{R \in \mathscr{A}[\ell]} h_P(x + R)$, then by comparing the divisors of the functions we obtain that (up to constant factor) $H_P = g_P^\ell$. By applying Lemma 4, we have $h_P(D_Q) = g_Q(Z_P)$. This gives :

$$\frac{\prod_{R \in \mathscr{A}[\ell]} h_P(Q_0 + R)}{\prod_{R \in \mathscr{A}[\ell]} h_P(R)} = \frac{g_Q^\ell(P_0)}{g_Q^\ell(0)} \frac{g_Q(P_0 - P)}{g_Q(P_0)}. \tag{19}$$

As $e_W(P, Q) = g_Q(P_0 - P)/g_Q(P_0)$ and taking into acount that $H_P = g_P^\ell$, we finally obtain:

$$e_W(P, Q) = \frac{g_P^\ell((Q_0) - (0))}{g_Q^\ell((P_0) - (0))} = \frac{f_{\ell D_P}(D_Q)}{f_{\ell D_Q}(D_P)},$$

where, for the second equality, we use the fact that $g_X^\ell = [\ell]^* f_{D_X}$ for $X = P, Q$.

It is also straightforward to show that the above formula for the Weil pairing depends only on the class of the cycles $(P) - (0)$ and $(Q) - (0)$ modulo equivalence and on the Chern class of the Theta divisor Θ. For more details, see [17, Section 6]. It is also easy to prove the bilinearity and non degeneracy as in [7].

Remark 1. In the elliptic case, one usually take the principal polarization coming from the neutral point (that is the point at infinity). Hence, one cannot evaluate $f_{\ell,P}$ at the cycle $(Q) - (0)$ since $f_{\ell,P}$ has a pole there. However, in our case we are taking a polarization coming from the Theta divisor, which usually does not contain 0. For the Jacobian of an hyperelliptic curve, the Theta divisor corresponds to degenerate divisors translated by a theta characteristic corresponding of a choice of odd roots of the Weierstrass function (see [26]), so it contains a point of two torsion that is usually different from 0.

In our case, we don't compute the Weil pairing using the principal polarization coming from the Theta divisor Θ, but we use the polarization coming from Θ_n. This mean that we will compute the n-th power of the standard Weil pairing, so we will assume that ℓ is prime to n in order to have a non-degenerate pairing in the rest of the paper. To compute this pairing, we simply replace the function $f_{\ell((P)-(0))}$ used in the definition of the Weil pairing by the function $f_{\ell,n,P}$ defined in Section 4.

5.2 The Tate pairing

In this section, we suppose for simplicity that $\mu_\ell \subset K$ and that $\mathscr{A}[\ell]$ is rational over K, the reader can consult [4] for the general case.

Let $\overline{K}$ be the algebraic closure of K and let $G = \mathrm{Gal}(\overline{K}/K)$ be the absolute Galois group of K. By taking the group cohomology long exact sequence associated to the Kummer exact sequence:

$$0 \to \mu_\ell \to \overline{K}^* \to \overline{K}^* \to 0,$$

and using the fact that $H^1(G, \overline{K}^*) = 0$ by Hilbert 90, we obtain an isomorphism

$$\delta_1 : K^*/K^{*\ell} \to \mathrm{H}^1(G, \mu_\ell) = \mathrm{Hom}(G, \mu_\ell).$$

In the same way, from the exact sequence

$$0 \to \mathscr{A}[\ell] \to \mathscr{A}(\overline{K}) \to \mathscr{A}(\overline{K}) \to 0,$$

we get a morphism

$$\delta_2 : \mathscr{A}(K)/[\ell]\mathscr{A}(K) \to \mathrm{Hom}(G, \mathscr{A}[\ell]).$$

There exists a bilinear application often referred to as the Tate pairing $e_T : \mathscr{A}[\ell] \times \mathscr{A}(K)/[\ell]\mathscr{A}(K) \to K^*/K^{*\ell}$ such that for $(P, Q) \in \mathscr{A}[\ell] \times \mathscr{A}(K)/[\ell]\mathscr{A}(K)$, $e_W(P, \delta_2(Q)) = \delta_1(e_T(P, Q))$.

It is well known in the case that $\mathscr{A}$ is an elliptic curve that one can compute the Tate pairing by taking any divisor D linearly equivalent to $(Q) - (0)$ and computing $e_T(P, Q) = f_{\ell,P}(D)$. This fact generalizes to any abelian variety with a principal polarisation.

Theorem 3. *Let $P, Q \in \mathscr{A}(\mathbb{F}_{q^k})$ such that P is a point of ℓ-torsion. Let D_P and D_Q be two cycles equivalent to $(P)-(0)$ and $(Q)-(0)$. Then we have $S(\ell D_P) = 0$ and let $f_{\ell D_P}$ be the corresponding function on $\mathscr{A}$. The (non reduced) Tate pairing is given by*

$$e_T(P, Q) = f_{\ell D_P}(D_Q). \tag{20}$$

Proof. Let $Q_0 \in \mathscr{A}(\overline{K})$ such that $\ell Q_0 = Q$. Following the definition of the connection morphism δ_2, we have $\delta_2(Q) = \mathfrak{f}$ where $\mathfrak{f} : G \to \mathscr{A}[\ell]$, $\sigma \mapsto Q_0^\sigma - Q_0$ is a co-cycle (in fact a morphism since $\mathscr{A}[\ell]$ is rational over K) representing an element of $H^1(G, \mu_\ell)$.

By definition of the Weil pairing, we have

$$e_W(Q_0^\sigma - Q_0, P) = \frac{g_P(Q_0)}{g_P(Q_0^\sigma)}. \tag{21}$$

On the other side, as $[\ell]^*(f_P) = c.(g_P)^\ell$ where $c \in \overline{K}$ is a constant, we have $\left(\frac{g_P(Q_0)}{g_P(0)}\right)^\ell = \frac{f_P(Q)}{f_P(0)}$. But then $\delta_1(f_P((Q) - (0)))$ is represented by the co-cycle $\mathfrak{g} : G \to \frac{g_P(Q_0)}{g_P(Q_0^\sigma)}$. Comparing this with the preceding equation concludes the proof.

Let $\mathscr{A}$ be an abelian variety over $\mathbb{F}_q$ a finite field of characteristic p. In order to use the preceding theory, we have to "lift" the abelian variety $\mathscr{A}$ over a field of characteristic 0. For this, we denote by $\mathfrak{R} = W(\mathbb{F}_q)$ the ring a Witt vectors with coefficients in $\mathbb{F}_q$ and by $\mathfrak{K}$ the quotient field of $\mathfrak{R}$. Denote by $\overline{\mathbb{F}}_q$ an algebraic closure of $\mathbb{F}_q$ and let π be the absolute Frobenius morphism.

An abelian scheme $\widetilde{\mathscr{A}}$ over $\mathfrak{R}$, the special fiber of which is $\mathscr{A}$ is said to be a lift of $\mathscr{A}$ over $\mathfrak{R}$. Of course such a lift is not unique in general. For the rest

of the section, we fix an embedding $\kappa : \mathfrak{R} \to \mathbb{C}$ so that we can consider $\mathfrak{R}$ as a subfield of $\mathbb{C}$ and we suppose that $\widetilde{\mathscr{A}}$ becomes an abelian variety defined over a number field K.

Let ℓ be a prime number different from p. We denote by χ_ℓ the characteristic polynomial of the Frobenius morphism acting on the ℓ-adic Tate module of $\mathscr{A}$. We recall (see [24, Theorem 4 p. 206]) that χ_ℓ is a degree $2g$ polynomial and if α_i are the roots of χ_ℓ then there is a permutation σ of $\{1, \ldots, 2g\}$ such that for $i = 1, \ldots, g$,

$$\alpha_{\sigma(i)} = q/\alpha_{\sigma(2i)}. \tag{22}$$

Denote by $\mathbb{G}_1 = \mathscr{A}[\ell] \cap \ker(\pi - 1)$ the Eigenspace of the Frobenius morphism acting on $\mathscr{A}[\ell]$. If ℓ divides the cardinal $\#\mathscr{A}(\mathbb{F}_q)$, then $\mathbb{G}_1$ is non trivial. In the same way, we let $\mathbb{G}_2 = \mathscr{A}[\ell] \cap \ker(\pi - [q])$ be the Eigenspace associated to the Eigenvalue q, if $\mathbb{G}_1$ is non trivial, then by (22), $\mathbb{G}_2$ is also non trivial. Denote by k the embedding degree of ℓ, that is the smallest integer such that $\ell | q^k - 1$ (so that $\mathbb{F}_{q^k}$ is the smallest extension of $\mathbb{F}_q$ containing μ_ℓ). We remark that $\mathbb{G}_2$ is defined over $\mathbb{F}_{q^k}$. By Hensel, we can lift $\mathscr{A}(\mathbb{F}_q)$ and $\mathbb{G}_2$ to $\widetilde{\mathscr{A}}$, as a set of points defined over $\mathfrak{R}$ and $W(\mathbb{F}_{q^k})$ respectively.

Reducing the Tate pairing on $\widetilde{\mathscr{A}}$ modulo p we get the Tate pairing as a non degenerate pairing on the right

$$e_T : \mathbb{G}_2 \times \mathscr{A}(\mathbb{F}_q)/\ell\mathscr{A}(\mathbb{F}_q) \to \mathbb{F}_{q^k}^*/\mathbb{F}_{q^k}^{*\ell}. \tag{23}$$

If $\mathscr{A}(\mathbb{F}_q)$ has no point of ℓ^2-torsion, we can identify $\mathscr{A}(\mathbb{F}_q)/\ell\mathscr{A}(\mathbb{F}_q)$ with $\mathbb{G}_1$, so that in particular we have that $\mathbb{G}_2$ has the same rank as $\mathbb{G}_1$.

Finally, if we assume that $\mathscr{A}(\mathbb{F}_{q^k})$ also has no points of ℓ^2-torsion, then by looking at the Tate pairing over $\mathbb{F}_{q^k}$, we get a non degenerate bilinear pairing

$$e_{T,r} : \mathscr{A}[\ell](\mathbb{F}_{q^k}) \times \mathscr{A}[\ell](\mathbb{F}_{q^k}) \to \mu_\ell \subset \mathbb{F}_{q^k}^* \tag{24}$$

by computing the reduced Tate pairing as $e_T(P, Q)^{\frac{q^k-1}{\ell}}$.

By definition, if P and Q are points in $\mathscr{A}[\ell](\mathbb{F}_{q^k})$, then the reduced Tate pairing is $e_W(P, \pi(Q') - Q')$ where Q' is any geometric point with $[\ell]Q' = Q$.

In practice, we compute the Tate pairing by using the formula of Theorem 3. We set $D_Q = (Q) - (0)$ so that $f_{\ell D_P}^n = f_{\ell,n,P}$ and compute $e_T(P, Q) = f_{\ell,n,P}(D_Q)$ if well defined. Otherwise, we replace D_Q by the equivalent cycle $(Q + R) - (R)$ where R is any point in $\mathscr{A}(\mathbb{F}_{q^k})$, such that this cycle is not in the support of the divisor associated to f_{ℓ,D_P}. Then we have

$$e_T(P, Q) = \frac{f_{\ell,n,P}(Q + R)}{f_{\ell,n,P}(R)}.$$

In order to check this directly, consider the cycle $(Q + R) - (Q) - (R) + (0)$. This cycle corresponds to a divisor linearly equivalent to 0; let g be a function associated to it. Lang's reciprocity yields $f_{\ell,P}((Q + R) - (Q) - (R) + (0)) = g(\ell(P) - \ell(0)) = (g(P)/g(0))^\ell \in \mathbb{F}_{q^k}^{*\ell}$. Hence $e_T(P, Q) = f_{\ell,P}((Q + R) - (R))$.

We remark as in Section 5.1 that since we use a non principal polarisation given by Θ_n, the Tate pairing that we compute is also equal to the usual Tate pairing to the power of n.

Remark 2. If $\mathscr{A}$ is the Jacobian of a curve C, then the points P and Q can be seen as divisors D_P and D_Q on C. Lichtenbaum showed in [19] that the Tate pairing can be computed directly by evaluating $f'_{\ell D_P}(D_Q)$ where $f'_{\ell D_P}$ is a function in the function field of C.

This has the advantage that the Lichtenbaum pairing only uses functions defined over the curve rather than the Jacobian. However, working directly on the curve has the drawback that twists of curves are not easy to relate explicitly to twists of the Jacobian, hence it is hard to generalize the twisted ate pairing in higher genus [12]. Since our point of view is to consider pairing on abelian varieties, it is easy for us to extend all results of [14] to higher dimension.

5.3 Computing the Weil and Tate pairings

Let $P \in \mathscr{A}[\ell]$, we are going to present two methods to compute the normalized functions $f_{\ell,n,P}/f_{\ell,n,P}(0)$. More precisely, suppose that we are given $P = (P_i)_{i\in Z(\overline{n})}$ and let $Q = (Q_i)_{i\in Z(\overline{n})}$ two geometric point in $\mathscr{A}$ embedded in $\mathbb{P}^{Z(\overline{n})}$ via the theta coordinates of level n. Let $z_P, z_Q \in \mathbb{C}^g$ be such that $P = (\theta_i(z_P))$ and $Q = (\theta_i(z_Q))$, we want to compute $f_{\ell,n,P}(z_Q)/f_{\ell,n,P}(0)$ from the knowledge of the homogeneous coordinates of the points P, Q and 0. By Section 5.1, this is sufficient to compute the Weil and Tate pairings (for the Weil pairing one has to repeat the procedure swapping P and Q).

Lemma 4, Lemma 5 and Corollary 1 give a first method we compute Weil and Tate pairings in a similar way to Miller's algorithm when the level n is divisible by 4. Take a random point $R \in \mathscr{A}(K)$. Let $Q \in \mathscr{A}[\ell]$; starting from the function $f_{1,n,P} = 1$, we can use relation (13) and Corollary 1 to compute by a square and multiply algorithm $f_{\ell,n,P}(Q+R)$ and $f_{\ell,n,P}(R)$. The algorithm terminates when $Q+R$ and R do not belongs to the poles of the functions $\mathfrak{f}_{\lambda,\mu,n,P}$ used in the computation. The Weil pairing can be computed in a similar manner. As explained in the introduction, all these results are valid when K is a finite field of characteristic different from 2 and give a probabilistic algorithm with expected probability arbitrarily close to 1 to output the result as the size of the K grows to infinity.

A second method is to use Proposition 3 directly. We will see in Propositions 5 and 6 that in fact we can always compute the Weil or Tate pairing.

Proposition 5. *Let $\widetilde{P+Q}$, $\widetilde{P}$, $\widetilde{Q}$ and $\widetilde{0}$ be affine lifts of $P+Q$, P, Q and 0, where P is a point of ℓ-torsion. Let λ_P^1, λ_P^0 be such that* $\mathrm{ScalarMult}(\ell, \widetilde{P+Q}, \widetilde{P}, \widetilde{Q}, \widetilde{0}) = \lambda_P^1 \widetilde{Q}$ *and* $\mathrm{ScalarMult}(\ell, \widetilde{P}, \widetilde{P}, \widetilde{0}, \widetilde{0}) = \lambda_P^0 \widetilde{0}$. *We have*

$$e_T(P,Q) = \frac{\lambda_P^0}{\lambda_P^1}. \tag{25}$$

If Q is also a point of ℓ-torsion, then we can define λ_Q^1, λ_Q^0 in a similar manner, and we have

$$e_W(P,Q) = \frac{\lambda_P^0}{\lambda_P^1}\frac{\lambda_Q^1}{\lambda_Q^0}. \tag{26}$$

Proof. Immediate by Proposition 3. See also [20].

Remark 3. Since P is a point of ℓ-torsion, the cycle $\ell(P) - (\ell P) - (\ell-1)(0)$ is equal to the cycle $(\ell+1)(P) - ([\ell+1]P) - \ell(0)$. In particular, we can also compute the Tate (and Weil) pairing by using $f_{\ell+1,n,P}$ rather than $f_{\ell,n,P}$. In the context of Proposition 5, this mean that we have $\text{ScalarMult}(\ell+1, \widetilde{P+Q}, \widetilde{P}, \widetilde{Q}, \widetilde{0}) = \lambda_P^1 \widetilde{P+Q}$, $\text{ScalarMult}(\ell+1, \widetilde{P}, \widetilde{P}, \widetilde{0}, \widetilde{0}) = \lambda_P^0 \widetilde{P}$ and $e_T(P,Q) = \lambda_P^0/\lambda_P^1$.

The fact that we can always compute the Weil and Tate pairings in Proposition 5 can be explained as follow. For $P \in \mathscr{A}[\ell]$ and $i \in Z(\overline{n})$, let $f_{\ell,n,P,i}$ be the function whose divisor is $\ell\tau_P^*\Theta_{n,i} - \tau_{\ell P}^*\Theta_{n,i} + (\ell-1)\Theta_{n,i}$ where we make explicit the dependency with respect to the divisor $\Theta_{n,i}$ corresponding to the coordinate θ_i. (As we have seen, $\Theta_{n,i}$ is equivalent to $\Theta_n = \Theta_{n,0}$ and is explicitly described as a translation of Θ_n by the point of n-torsion corresponding to i via the action of the Theta group).

Then the Tate pairing can be given following Lemma 3 by:

$$e_T(P,Q) = f_{\ell,n,P,i}((Q)-(0)) = \frac{\theta_i(z_Q)}{\theta_i(\ell z_P + z_Q)}\frac{\theta_i(\ell z_P)}{\theta_i(0)}$$

if this equation is well defined. If not, we could always replace the cycle $(Q)-(0)$ by an equivalent cycle, but we can also replace $\Theta_{n,i}$ by the equivalent divisor $\Theta_{n,j}$, that is compute the Tate pairing as

$$e_T(P,Q) = f_{\ell,n,P,j}((Q)-(0)) = \frac{\theta_j(z_Q)}{\theta_j(\ell z_P + z_Q)}\frac{\theta_j(\ell z_P)}{\theta_j(0)}$$

if this is well defined.

But the first term correspond to the projective factor $1/\lambda_P^1$ and the second to λ_P^0 in the notations of Proposition 5. This means that we can also compute the Tate pairing as

$$e_T(P,Q) = \frac{\theta_i(z_Q)}{\theta_i(\ell z_P + z_Q)}\frac{\theta_j(\ell z_P)}{\theta_j(0)} \tag{27}$$

and we can always find $i, j \in Z(\overline{n})$ such that this expression is well defined.

It is well known (see for instance [12]) that in the case that (0) is in the pole of the function $f_{\ell,n,P}$ then the Tate pairing can be defined as $f_{\ell,n,P}(Q)/c$ where c correspond to the leading coefficient of $f_{\ell,n,P}$ in the completion of Θ_n along $\mathscr{A}$. Equation (27) can be seen as a version of this since the leading coefficient of $f_{\ell,n,P,i}$ is the same as $f_{\ell,n,P,j}$ and the latter can be obtained as $\frac{\theta_j(\ell z_P)}{\theta_j(0)}$ (up to a ℓ^{th}-power) when (0) is not in the pole of $\Theta_{n,j}$.

But the fact that we can always compute the Tate pairing can be seen as a misfeature in term of complexity. It means that computing $\mathrm{ScalarMult}(\ell, \widetilde{P+Q}, \widetilde{P}, \widetilde{Q}, \widetilde{0}) = \lambda_P^1 \widetilde{Q}$ amount to computing the n^g Miller's functions $f_{\ell,n,P,j}((Q)-(0))$. It would be interesting to know if we could compute only one such function $f_{\ell,n,P,0}((Q)-(0))$ from $\mathrm{ScalarMult}(\ell, \widetilde{P}, \widetilde{P}, \widetilde{0}, \widetilde{0})$ and the coordinates of Q (and eventually $P+Q$).

We can always use Proposition 5 when $4 \mid n$ to compute the Weil and Tate pairings. In level $n = 2$, we need the point $P + Q$. For instance, we can work at first in level 4 and then switch to level 2 (using the duplication or the isogeny formula) to compute the pairing once we have $P + Q$.

In general, if we start from P and Q in level 2, we can only compute the symmetric Weil (or Tate) pairing $e_W(P,Q) + e_W(-P,Q)$ as defined in [20, Section 5.2]. To do this, we compute $P + Q$ in the algebra $\mathfrak{A}$ from Section 3, and do all the computations in this algebra of dimension 2. In practice, it seems faster to take a square root to fix a choice of $P + Q$ in the field of definition K and do all the remaining computation in K rather than in $\mathfrak{A}$.

In a sense, the second method, by computing differential additions on affine lifts, can be thought of as using the definition of the Weil pairing as the commutator pairing of the theta group [22] (or in the analytic version, as the symplectic pairing induced by the Riemann form [25]). But if we unravel the first version using the classical Miller algorithm with theta functions, by looking at Lemma 3, we see that Proposition 2 actually is just another way to compute an affine scalar multiplication.

Proposition 6. *Let $\widetilde{P+Q}$, $\widetilde{P}$, $\widetilde{Q}$ and $\widetilde{0}$ be affine lifts of $P+Q$, P,Q and 0, where P is a point of ℓ-torsion. Use the following algorithm to compute lifts $\widetilde{\ell P}$ and $\widetilde{Q+\ell P}$ by a double and add method:*

Input: *Affine lifts $\widetilde{\lambda P}$ and $\widetilde{Q+\lambda P}$.*

Double: *(at each step.) Compute $\widetilde{2\lambda P}$ and $\widetilde{Q+2\lambda P}$ with two differential additions.*

Add: *(only if the current bit of ℓ is one.) From the points $2\lambda P$, P, $Q+P$ and $Q+2\lambda P$ use a compatible addition to compute the projective point $(2\lambda+1)P$ (if $4 \mid n$ one can of course use a normal addition directly); and take an arbitrary lift. From the affine lifts $\widetilde{2\lambda P}$, $\widetilde{Q}$, $\widetilde{P}$, $\widetilde{Q+P}$, $\widetilde{(2\lambda+1)P}$ and $\widetilde{Q+2\lambda P}$, do a three way addition to compute an affine lift $\widetilde{Q+(2\lambda+1)P}$.*

Let λ_P^1, λ_P^0 be such that $\widetilde{Q+\ell P} = \lambda_P^1 \widetilde{Q}$ and $\widetilde{\ell P} = \lambda_P^0 \widetilde{0}$, then we have

$$e_T(P,Q) = \frac{\lambda_P^0}{\lambda_P^1}. \tag{28}$$

And a similar result holds for the Weil pairing. In particular, we can always compute the Tate and Weil pairing when $4 \mid n$.

Proof. This is a direct application of Proposition 2 and Lemma 5. Indeed, by Corollary 1, the result does not depend on the choice of lift of $(2\lambda+1)P$ done after each addition step.

Compared to the method of Proposition 5, which uses three differential additions at each step (whatever the current bit of ℓ is), the method of Proposition 6 only uses two differential addition at each "doubling" step. However, it requires a compatible addition and a three way addition at each "addition" step. Still, it may be worthwhile to use when the hamming weight of ℓ is small, especially combining with a NAF method.

6 Ate and symmetric ate pairings

In this section, we give a generalisation of ate pairings to all abelian varieties. We keep the notations of Section 5.2.

For the Tate pairing, one usually take $P \in \mathbb{G}_1$ and $Q \in \mathbb{G}_2$ for efficiency reasons. However, if $P \in \mathbb{G}_2(\mathbb{F}_{q^k})$, then

$$\pi(f_{\ell,n,P}) = f_{\ell,n,qP}. \tag{29}$$

(See [14, Lemma 3]). The idea of the ate pairing is to switch the role of P and Q (that is take $P \in \mathbb{G}_2$ and $Q \in \mathbb{G}_1$) and use this relation in order to decrease the number of iterations in the computation of the Tate pairing.

For this we take $\mu \equiv q \mod \ell$. We remark that as $\ell | q^k - 1$, we have $\ell | \mu^k - 1$ and we put $m = \frac{\mu^k - 1}{\ell}$. Let n be the level of our theta functions. We compute:

$$e_T(P,Q)^m = f_{m\ell,n,P}((Q) - (0)) = f_{\mu^k - 1,n,P}((Q) - (0)) = f_{\mu^k,n,P}((Q) - (0)) \tag{30}$$

The first equality is a consequence of Lemma 4 and the second one comes from the definition of $f_{\mu^k-1,n,P}$.

Then by a repeated use of Lemma 4 and (29) we get

$$\begin{aligned} f_{\mu^k,n,P}((Q) - (0)) &= \prod_{i=1}^{k-1} f^{\mu^i}_{\mu,n,\mu^{k-1-i}P}((Q) - (0)) \\ &= \prod_{i=1}^{k-1} \pi^{k-1-i}(f^{\mu^i}_{\mu,n,P}((Q) - (0))). \end{aligned}$$

We can define the (n-power of the usual) ate paring $e_A : \mathbb{G}_2 \times \mathbb{G}_1 \to \mathbb{F}^*_{q^k}/\mathbb{F}^{*\ell}_{q^k}$ by

$$e_A(P,Q) = f_{\mu,n,P}((Q) - (0)).$$

By the equations above we have,

$$e_A(P,Q)^c = e_T(P,Q)^m \tag{31}$$

where $c = \sum_{i=0}^{k-1} \mu^i q^{k-1-i}$.

The reduced ate pairing is then equal to the reduced Tate pairing to the power of m, so this pairing is non degenerate if ℓ does not divide $m\frac{q^k-1}{\ell}$. A trick to reduce m is to divide it by $q^k - 1 \wedge \mu^k - 1$. Indeed, the (reduced) $h\ell$ Tate pairing computed on a point of ℓ-torsion is equal to its (reduced) ℓ-Tate pairing.

Remark 4. If $\mathscr{A}$ is an elliptic curve, let t be the trace of the Frobenius morphism. If $\ell = q+1-t$ is a prime number, we can take $\mu = 1-t$ which by the Weil bound is a $O(\sqrt{q})$. In this case, the expected number of iteration needed to compute the ate pairing is less than half of those required for the computation of the Tate pairing so that we can expect a speed up.

If $\mathscr{A}$ is an abelian variety of dimension greater or equal to 2, we have $\mu \leqslant q$ while $\sharp\mathscr{A}(\mathbb{F}_q) = O(q^g)$, so usually ℓ is greater than q. In this case we gain a g-fold speedup in the number of iteration to compute the Miller function.

Note that in the case $\mu = q$ (for instance when $g \geqslant 2$), then $f_{\mu,n,P}((Q)-(0))$ is already reduced so there is no need for the final exponentiation, and we can replace the cycle $(Q)-(0)$ by any equivalent cycle [12].

An algorithm to compute e_A is provided by Proposition 3. Before the final exponentiation, we need to compute $\mathrm{ScalarMult}(\mu, \widetilde{P+Q}, \widetilde{P}, \widetilde{Q}, \widetilde{0})$, $\mathrm{ScalarMult}(\mu, \widetilde{P}, \widetilde{P}, \widetilde{0}, \widetilde{0})$ and $\left(\frac{(\widetilde{P+Q})_i.(\widetilde{0})_i}{(\widetilde{P})_i.(\widetilde{Q})_i}\right)^{\mu}$. As a consequence, these calculations can be done at the cost of $O(\log \mu)$ iterations using a fixed number of operations in the field $\mathbb{F}_{q^k}$. Explicitly:

Proposition 7. *Let $\widetilde{P+Q}$, $\widetilde{P}$, $\widetilde{Q}$ and $\widetilde{0}$ be affine lifts of $P+Q$, P,Q and 0, where $P \in \mathbb{G}_2$ and $Q \in \mathbb{G}_1$. Let λ_P^1, λ_P^0 be such that $\mathrm{ScalarMult}(\mu, \widetilde{P+Q}, \widetilde{P}, \widetilde{Q}, \widetilde{0}) = \lambda_P^1 \pi^\mu(\widetilde{P+Q})$ and $\mathrm{ScalarMult}(\mu, \widetilde{P}, \widetilde{P}, \widetilde{0}, \widetilde{0}) = \lambda_Q^P \pi^\mu(\widetilde{P})$. We have*

$$e_A(P,Q) = \frac{\lambda_P^0}{\lambda_P^1}. \tag{32}$$

Remark 5. By looking at the differential additions, we can recover the power in which the (non reduced) Tate pairing and ate pairing correspond.

Recall that $\mu^k - 1 = m\ell$. Suppose that $\widetilde{\ell P+Q} = \alpha\widetilde{Q}$ and $\widetilde{\ell P} = \beta\widetilde{0}$, so that α/β gives the Tate pairing. Then $\widetilde{m\ell P+Q} = \beta^m\widetilde{Q}$ and $\widetilde{\mu^k P+Q} = \beta^m\widetilde{P+Q}$, and $\widetilde{\mu^k P} = \alpha^m\widetilde{P}$.

For $1 \leqslant i \leqslant k$, let $\widetilde{\mu^i P+Q} = \gamma_i\pi^i(\widetilde{P+Q})$ and $\widetilde{\mu^i P} = \delta_i\pi^i(\widetilde{P})$, so that $e_A(P,Q) = \gamma_1/\delta_1$ and $e_T(P,Q)^m = \gamma_m/\delta_m$.

If we write $\widetilde{\mu^i P+Q} = \gamma_i'\pi(\widetilde{\mu^{i-1}P+Q})$ and $\widetilde{\mu^i P} = \delta_i'\pi(\widetilde{\mu^{i-1}P})$, then since π commutes with differential additions we get by Lemma 1

$$\widetilde{\mu^{i+1}P+Q} = (\delta_i')^{\mu(\mu-1)}(\gamma_i')^\mu\pi(\widetilde{\mu^i P+Q})$$
$$\widetilde{\mu^{i+1}P} = (\delta_i')^{\mu^2}\pi(\widetilde{\mu^i P}).$$

By a trivial recursion, we have

$$\frac{\gamma_{i+1}}{\delta_{i+1}} = \left(\frac{\gamma_1}{\delta_1}\right)^{\mu^i}\left(\frac{\gamma_i}{\delta_i}\right)^q,$$

which give back $e_T(P,Q)^m = e_A(P,Q)^c$, with $c = \sum_{i=0}^{k-1}\mu^i q^{k-1-i}$.

Finally, we can apply the same trick to the symmetric Tate pairing of [20] to obtain in the same way a symmetric ate pairing on Kummer varieties.

6.1 Twisted ate pairing

We have remarked that in the ate pairing, the length of the Miller loop is less than in a regular Tate pairing. This comes at a cost however. In the ate pairing, we need to compute $f_{\mu,n,P}$, where $P \in \mathbb{G}_2$ lives in the "big field" $\mathbb{F}_{q^k}$. When doing the Tate pairing, one can instead compute $f_{\ell,n,P}$, where $P \in \mathbb{G}_1$ lives in the "small field" $\mathbb{F}_q$; in this case only the evaluation $f_{\ell,n,P}(Q)$ is done in $\mathbb{F}_{q^k}$.

In the supersingular case, since the action of the Verschiebung $\hat{\pi}$ is inseparable and acts by multiplication by q on $\mathbb{G}_1$ and 1 on $\mathbb{G}_2$, we can define in a similar manner the eta pairing as $f_{\mu,n,P}(Q)$ where $P \in \mathbb{G}_1$ and $Q \in \mathbb{G}_2$ like the Tate pairing [14].

In the ordinary case, to alleviate this problem, one usually combines twists with the ate pairing. Suppose that there exists a twist of degree $d \mid k$, and let $e = k/d$. Then this twist correspond to a d^{th}-root of unity ζ in the endomorphism ring. We have $\pi^k - 1 = (-1)^{d-1} \prod_{i=0}^{d-1} \zeta^i \pi^e - 1$. Now since ζ is of order d, and that d is prime to ℓ (since $k < \ell$), we have that $\mathscr{A}[\ell](\mathbb{F}_{q^k}) = \bigoplus \mathrm{Ker}(\zeta^i \pi^e - 1)[\ell])$. If $\mathbb{G}_1$ is of rank 1, then $\mathbb{G}_2 \subset \mathscr{A}[\ell](\mathbb{F}_{q^k})$ is also, so there is a unique twist $\mathscr{A}'$ of $\mathscr{A}$ over $\mathbb{F}_{q^e}$, corresponding to ζ^i, such that $\mathbb{G}_2 \xrightarrow{\sim} \mathscr{A}'[\ell](\mathbb{F}_{q^e})$.

Mapping a point of $\mathbb{G}_2$ via this twist, we get a point rational over $\mathbb{F}_{q^e}$ (while the points in $\mathbb{G}_1$ are defined over $\mathbb{F}_{q^k}$ on the twist). So one can use the twist map to compute $f_{\mu,n,P}$ in the twist and come back to evaluate at Q. This has the advantage that in the twist, P only lives in the extension $\mathbb{F}_{q^e}$. However, to apply this idea to our setting, we need a rational theta structure on both the abelian variety $\mathscr{A}$ and its twist, which is impossible since the theta structure rigidify the moduli stack of abelian varieties by [23] (at least in level 4).

One can instead compute everything in the twist, as explained in [6]. In this case, we only need a rational theta structure on the twist, and P is defined over $\mathbb{F}_{q^e}$ while Q is defined over $\mathbb{F}_{q^k}$. (If P' and Q' are the twisted points, then $e_A(P', Q')^d = e_A(P, Q)^d$. In [6], they have a finer result for elliptic curves by analysing the updates of Miller functions).

Another method is to compute the twisted ate pairing $e_{A'}(P, Q) = f_{\mu^e,n,P}((Q) - (0))$ [14], where $P \in \mathbb{G}_1$ and $Q \in \mathbb{G}_2$ so that the computation of $f_{\mu^e,n,P}$ can be done in the smaller field, but at the cost of a larger loop than the ate pairing. The idea behind the twisted ate pairing is that pulling back the action of the Frobenius of $\mathbb{F}_{q^e}$ of the twist $\mathscr{A}'$ to $\mathscr{A}$, we get that $\zeta^i \pi^e$ acts as an inseparable endomorphism ψ of degree q^e on $\mathscr{A}$, with $\psi(P) = [q^e]P$ and $\psi(Q) = Q$ when $P \in \mathbb{G}_1$, $Q \in \mathbb{G}_2$.

7 Optimal and symmetric optimal pairings

We recall the following proposition from [29, Theorem 1], stated for elliptic curves but which adapt easily to abelian varieties, as in Section 6 (from where we take the notations $\mathbb{G}_1$ and $\mathbb{G}_2$).

Proposition 8. *Let $\lambda = m\ell = \sum c_i q^i$ be a multiple of ℓ (and such that $\ell \nmid m$). The pairing*

$$a_\lambda : \ \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mu_\ell$$
$$(P, Q) \longmapsto \left(\prod_i f_{c_i,n,Q}(P)^{q^i} \prod_i \mathfrak{f}_{\sum_{j>i} c_j q^j, c_i q^i, n, Q}(P) \right)^{(q^k-1)/\ell}$$

is non degenerate when

$$mdq^{d-1} \not\equiv \frac{q^k - 1}{r} \sum_i i c_i q^{i-1} \mod \ell.$$

The idea is then to find a multiplie λ such that the coefficients c_i are small. More precisely, since $\varphi_k(q) = 0 \mod \ell$, one can use LLL to find a small relation among the powers $q, q^2, \ldots, q^{\varphi(k)-1}$. The discussion in [29, Section 3.3] shows that we can expect to find λ such that $c_i \approx \ell^{1/\varphi(k)}$.

Algorithm 7: Optimal ate

input : $P \in \mathbb{G}_1$, $Q \in \mathbb{G}_2$, $\lambda = m\ell = \sum_{i<k} c_i q^i$.
output : $a_\lambda(P, Q)$

1 Take affine lifts $\widetilde{P}$, $\widetilde{Q}$ and $\widetilde{P+Q}$;
2 **for** $i = k-1, \ldots, i = 0$ **do**
3 Compute $\widetilde{P} + c_i\widetilde{Q}$ and $c_i\widetilde{Q}$ using ScalarMult;
4 Apply the Frobenius endomorphism to obtain $\widetilde{P} + c_i q^i \widetilde{Q}$ and $c_i q^i \widetilde{Q}$;
5 Compute $c_i q^i \widetilde{Q} + \sum_{j>i} c_j q^j \widetilde{Q}$ (up to a constant, using NormalAdd) and then use the extended Riemann relations from Proposition 2 to compute $\widetilde{P} + c_i q^i \widetilde{Q} + \sum_{j>i} c_j q^j \widetilde{Q}$ (up to the same constant);
6 Find the constants C_0 and C_1 such that we have $\lambda\widetilde{Q} = C_0 * \widetilde{Q}$ and $\widetilde{P} + \lambda\widetilde{Q} = C_1 * \widetilde{P}$;

Return : $(C_1/C_0)^{\frac{q^k-1}{\ell}}$

We can easily compute such a pairing by using the results of Section 4. In Algorithm 7 we give the corresponding algorithm when $4 \mid n$. When $n = 2$, as in Section 5 we either compute $\widetilde{P+Q}$ by working in $\mathfrak{A}$ all the time, or by taking a square root to get back in K. All the steps of the algorithm are then the same, except for Step 5 where we use CompatAdd on $c_i q^i \widetilde{Q}$, $\sum_{j>i} c_j q^j \widetilde{Q}$, $P + c_i q^i \widetilde{Q}$, $P + \sum_{j>i} c_j q^j \widetilde{Q}$ to compute $c_i q^i \widetilde{Q} + \sum_{j>i} c_j q^j \widetilde{Q}$.

The extension to pairing lattices [13] is also straightforward.

8 Performance comparisons

For the performance analysis, we only consider the case of level $n = 2$ since it gives the fastest representation and arithmetic. To compute the pairing between P and

Q, we have first to compute the coordinates of the point $P+Q$. This can be done in two ways: either we work with a system of coordinates where we can perform group law addition, like theta functions of level 4 or Mumford coordinates if $\mathscr{A}$ is the Jacobian of an hyperelliptic curve. After computing $P+Q$ in this system of coordinates, we can convert P, Q and $P+Q$ to theta coordinates of level 2 (if working with theta functions of level four, we can convert to level 2 by using the duplication formula, or by using the isogeny formula, which is essentially free since the level two coordinates of the isogenous points are a subset of the level four coordinates of the original points). If we only have the points P and Q in theta coordinates of level 2, then we can use the results of Section 3.3 to compute $P \pm Q$ either by computing with a degree 2 an algebra over the field of definition of the points or taking a square root (if q is congruent to 3 modulo 4 this can be done by an exponentiation). In this case, we can only compute the symmetric pairing. In all cases, to be able to work with theta coordinates of level 2, we need the theta null point of level 2 to be rational.

The main loop of the different algorithms rest in ScalarMult to compute the functions $f_{\lambda,n,P}$. Each step of the loop will then consists in a doubling, and two differential additions with the same point Q as difference. In the optimal ate pairings, there is some additional computations required to obtain the functions $\mathfrak{f}_{\lambda,\mu,n,P}$, but there is at most $k-1$ such functions to compute, while the length of the loop is in $\log(r^{1/\varphi(k)})$, so we can safely ignore their contribution to the running time.

This justify that we focus on the complexity of one step in the evaluation of the Miller function $\mathfrak{f}_{\lambda,\mu,n,P}(Q)$, according to whether P and Q is defined over the "big field" $\mathbb{F}_{q^k}$ or the base field $\mathbb{F}_q$. We denote by $\mathbf{M}$ a multiplication in $\mathbb{F}_{q^k}$, $\mathbf{S}$ a square in $\mathbb{F}_{q^k}$, $\mathbf{m}$ a multiplication by a "constant" in $\mathbb{F}_{q^k}$ coming from the coordinate of P, Q or $P+Q$. The corresponding operations over $\mathbb{F}_q$ are denoted by M, S and m respectively, and we denote by m_0 a multiplication by a constant depending only on the abelian variety (that comes from the theta null point). Lastly, we let M, m and $\mathsf{m_0}$ be the multiplication between an element of $\mathbb{F}_{q^k}$ and an element of $\mathbb{F}_q$ (respectively a constant in $\mathbb{F}_q$ depending only on the coordinates of P, Q or $P+Q$, and a constant in $\mathbb{F}_q$ depending only on the theta null point).

We first focus on the case where P and Q are defined over the big field. This is not the case in cryptography where we take a point in $\mathbb{F}_q$ to speed up the computations, but this is unavoidable in situation where one has to compute the Weil pairing between points of ℓ-torsion (for instance, to get a symplectic basis of the ℓ-torsion).

In dimension 1, one step of the "Miller loop" using ScalarMult as in Proposition 5 is given in Algorithm 8 (for the Weil pairing, we will need two such loops). We first note that we can always choose the lifts $\widetilde{P}$, $\widetilde{Q}$ and $\widetilde{P+Q}$ so that $x_P = 1, x_Q = 1, x_{P+Q} = 1$. We see that a step takes $5\mathbf{M} + 2\mathbf{m} + 7\mathbf{S} + 2\mathsf{m_0}$ in dimension 1. We note that we can get a $1\mathbf{S} + 1\mathsf{m_0} - 1\mathbf{M}$ trade-off by computing $Z_n = \beta(x_n^2 - z_n^2)$ as $Z_n = \frac{A}{B}(x_n^2 - z_n^2)^2$. In dimension 2, as in [20] we obtain that the cost of one step is $11\mathbf{M} + 6\mathbf{m} + 13\mathbf{S} + 6\mathsf{m_0}$. Similar to the dimension one case, there is a possible trade-off of $3\mathbf{S} + 3\mathsf{m_0} - 3\mathbf{M}$.

Algorithm 8: One step of the differential addition Miller loop

Input $nQ = (x_n, z_n)$; $(n+1)Q = (x_{n+1}, z_{n+1})$, $(n+1)Q + P = (x'_{n+1}, z'_{n+1})$.
Output $2nQ = (x_{2n}, z_{2n})$; $(2n+1)Q = (x_{2n+1}, z_{2n+1})$;
$(2n+1)Q + P = (x'_{2n+1}, z'_{2n+1})$.

1. $\alpha = (x_n^2 + z_n^2)$; $\beta = \frac{A}{B}(x_n^2 - z_n^2)$.
2. $X_n = \alpha^2$; $X_{n+1} = \alpha(x_{n+1}^2 + z_{n+1}^2)$; $X'_{n+1} = \alpha(x'^2_{n+1} + z'^2_{n+1})$;
3. $Z_n = \beta(x_n^2 - z_n^2)$; $Z_{n+1} = \beta(x_{n+1}^2 - z_{n+1}^2)$; $Z'_{n+1} = \beta(x'^2_{n+1} + z'^2_{n+1})$;
4. $x_{2n} = X_n + Z_n$; $x_{2n+1} = (X_{n+1} + Z_{n+1})/x_P$; $x'_{2n+1} = (X'_{n+1} + Z'_{n+1})/x_Q$;
5. $z_{2n} = \frac{a}{b}(X_n - Z_n)$; $z_{2n+1} = (X_{n+1} - Z_{n+1})/z_p$; $z'_{2n+1} = (X'_{n+1} - Z'_{n+1})/z_Q$;
6. Output (x_{2n}, z_{2n}); (x_{2n+1}, z_{2n+1}); (x'_{2n+1}, z'_{2n+1}).

We note that the pairing algorithm relying on theta functions, despite being a very generic algorithm available for all abelian varieties is actually pretty fast in small dimension. As a comparison, just doubling a point using the fastest known arithmetic for Mumford projective coordinates on a Jacobian of a curve of genus 2 (which is a necessary step for the Miller algorithm) already takes $33\mathbf{M} + 7\mathbf{S} + 1\mathsf{m_0}$ [18]! For elliptic curve, just the Doubling step in the Miller loop of a point on an Edwards curve takes $9\mathbf{M} + 7\mathbf{S} + 2\mathsf{m_0}$ (since there is no denominator elimination to compute the Weil pairing). Also, as already noted, if the Hamming weight of ℓ is small, the algorithm from Proposition 6 could be faster than the one from Proposition 5.

Now for pairings used in cryptography, one can usually choose which type of points to compute with. For the Tate pairing, this means that P is defined over $\mathbb{F}_q$ while Q will is defined over $\mathbb{F}_{q^k}$. Moreover, when the embedding degree k is even one can do denominator elimination, and in genus 2 one can take for Q a degenerate divisor. Table 2 shows the comparison between the algorithm from Proposition 5 with the usual Miller algorithm, with or without denominator elimination. More precisely, in genus 1 the cost for one step of the Miller loop using theta coordinates is $1\mathbf{m} + 2\mathbf{S} + 2\mathsf{M} + 3M + 1m + 5S + 2m_0$ (one can also use the $1S+1m_0-1M$ trade-off). In genus 2, the cost is $3\mathbf{m}+4\mathbf{S}+4\mathsf{M}+7M+3m+9S+6m_0$ (or with the $3S + 3m_0 - 3M$ trade-off). Here, the algorithm from Proposition 6 is not interesting because it only reduces operations in the smaller field for Doubling, while adding operations in the bigger field for each Addition. As a comparison, a pairing step with Edwards coordinates using denominator elimination costs $1\mathbf{M}+1\mathbf{S}+1\mathsf{M}+6M+5S+2m_0$ for each Doubling, and $1\mathbf{M}+1\mathbf{S}+1\mathsf{M}+12M+1m_0$ for each Addition. We see that for the Tate pairing, our algorithm performs poorly because while we do less operations in the small field, we compute more in the big field. In genus 1, Table 2 shows that this is because we do not have denominator elimination.

For the ate (and optimal ate) pairing, the Miller loop is shortened, but P lives in $\mathbb{F}_{q^k}$ while Q lives in $\mathbb{F}_q$. Since most operations take place in the big field, we expect our algorithm to be competitive, since it does less computations overall when all points are in the big field. For instance in dimension 1, one step costs

$5\mathbf{M} + 1\mathbf{m} + 7\mathbf{S} + 1\mathsf{m} + 2\mathsf{m}_0$, and in dimension 2 $11\mathbf{M} + 3\mathbf{m} + 13\mathbf{S} + 3\mathsf{m} + 6\mathsf{m}_0$ (with the possible $1\mathbf{S} + 1\mathsf{m}_0 - 1\mathbf{M}$ and $3\mathbf{S} + 3\mathsf{m}_0 - 3\mathbf{M}$ trade-offs respectively). In general, the cost of doing more computations in the bigger field offsets the reduced loop, so one usually use ate pairings in presence of twists. If there is a twist of degree $d \mid k$, then there is a twist that will send P to a point in $\mathbb{F}_{q^e}$ (where $e = k/d$) and Q in $\mathbb{F}_{q^k}$. Computing the pairing entirely on the twist then costs about the same as the Tate pairing, but with operations in $\mathbb{F}_q$ replaced by operations in $\mathbb{F}_{q^e}$ (see [6]), but with still the same loop length gain as the ate pairing. Depending on the size of e and k, our algorithm may be competitive in this case (and it could extend the range where the ate pairing is faster than the Tate pairing).

In dimension 2, according to [12] the ate pairing using affine Mumford coordinates costs $1\mathbf{I} + 29\mathbf{M} + 5\mathbf{S} + 7\mathsf{M}$ for an addition, and $1\mathbf{I} + 29\mathbf{M} + 9\mathbf{S} + 7\mathsf{M}$ for a doubling, where $\mathbf{I}$ denotes the cost of an affine inversion in $\mathbb{F}_{q^k}$. Even when using degenerate divisors, the cost is still of $1\mathbf{I} + 27\mathbf{M} + 3\mathbf{S} + 4\mathsf{M}$ for an addition and $1\mathbf{I} + 27\mathbf{M} + 7\mathbf{S} + 4\mathsf{M}$ for a doubling, so our formulas are way faster. In dimension 3, our formula for the ate pairing (without any optimization as settings some projective coordinates to one) will be in $32\mathbf{M} + 24\mathbf{S} + 8\mathsf{M} + 8\mathsf{m} + 16\mathsf{m}_0$, which is faster than only a doubling step using degenerates divisor using affine Mumford coordinates.

In genus 1, we expect the optimal ate pairing to gain a factor of $\varphi(k)$ in the loop length where k is the embedding degree. We would prefer k to be uneven, since this will increase the size of $\varphi(k)$. But the denominator elimination trick only work in the even case (even if there are some adaptations available in the uneven case). In dimension $g > 1$, we expect to have $\ell > q$, so the reduction μ mod ℓ in Section 6 is equal to q. In this case the ate pairing is already reduced [12] so there is no need for the final exponentiation. This means that one can't use a denominator elimination. But since in our algorithm we don't use denominator elimination anyway, these cases are actually favorable for the algorithms presented in this paper.

		Miller		Theta coordinates
		Doubling	Addition	One step
$g = 1$	k even	$1\mathbf{M} + 1\mathbf{S} + 1\mathsf{M}$	$1\mathbf{M} + 1\mathsf{M}$	$1\mathbf{m} + 2\mathbf{S} + 2\mathsf{M}$
	k odd	$2\mathbf{M} + 2\mathbf{S} + 1\mathsf{M}$	$2\mathbf{M} + 1\mathsf{M}$	
$g = 2$	Q degenerate + denominator elimination	$1\mathbf{M} + 1\mathbf{S} + 3\mathsf{M}$	$1\mathbf{M} + 3\mathsf{M}$	$3\mathbf{m} + 4\mathbf{S} + 4\mathsf{M}$
	General case	$2\mathbf{M} + 2\mathbf{S} + 18\mathsf{M}$	$2\mathbf{M} + 18\mathsf{M}$	

Table 1. $P \in A[\ell](\mathbb{F}_q)$, $Q \in A[\ell](\mathbb{F}_{q^k})$ (counting only operations in $\mathbb{F}_{q^k}$).

9 Conclusion

The main purpose of the paper is to give an algorithm to compute all known pairings (in particular the optimal ate pairing) on *any* abelian variety represented by theta functions (of even level). For efficiency reason, a particular focus was given on the level 2 case, which correspond to Kummer varieties rather than abelian varieties which lead to some difficulties. As seen in Section 8, while generic, the algorithm is surprisingly fast in lower dimension. To be truly competitive with the best pairing algorithms used in cryptography, it would be interesting to know if a denominator elimination is possible. It would also be worthwhile to investigate the case of degenerate divisors to speed up the pairing computation.

According to a recent paper [3], the arithmetic in dimension 2 can be faster than in dimension 1 (for a 128-bit security level). To achieve this speed, they use the representation given by level 2 theta functions. The ability to compute optimal pairings with such functions as explained in this paper, definitively show that the dimension 2 case is worth studying for pairings applications!

References

1. Christophe Arène, Tanja Lange, Michael Naehrig, and Christophe Ritzenthaler. Faster computation of the Tate pairing. *J. Number Theory*, 131(5):842–857, 2011. With supplementary material available online.
2. Paulo S. L. M. Barreto, Steven D. Galbraith, Colm Ó' hÉigeartaigh, and Michael Scott. Efficient pairing computation on supersingular abelian varieties. *Des. Codes Cryptogr.*, 42(3):239–271, 2007.
3. Joppe W Bos, Craig Costello, Huseyin Hisil, and Kristin Lauter. Two is greater than one. Technical report, Cryptology ePrint Archive, Report 2012/670, 2012. Available at: http://eprint. iacr. org/2012/670.
4. Henri Cohen, Gerhard Frey, Roberto Avanzi, Christophe Doche, Tanja Lange, Kim Nguyen, and Frederik Vercauteren, editors. *Handbook of elliptic and hyperelliptic curve cryptography*. Discrete Mathematics and its Applications (Boca Raton). Chapman & Hall/CRC, Boca Raton, FL, 2006.
5. Romain Cosset and Damien Robert. An algorithm for computing (ℓ, ℓ)-isogenies in polynomial time on jacobians of hyperelliptic curves of genus 2. HAL: hal-00578991, eprint: 2011/143, 03 2011.
6. Craig Costello, Tanja Lange, and Michael Naehrig. Faster pairing computations on curves with high-degree twists. *Public Key Cryptography–PKC 2010*, pages 224–242, 2010.
7. Andreas Enge. Bilinear pairings on elliptic curves. HAL: , 2012.
8. Steven D. Galbraith, Florian Hess, and Frederik Vercauteren. Hyperelliptic pairings. In *Pairing-based cryptography—Pairing 2007*, volume 4575 of *Lecture Notes in Comput. Sci.*, pages 108–131. Springer, Berlin, 2007.
9. Steven D Galbraith and Xibin Lin. Computing pairings using x-coordinates only. *Designs, Codes and Cryptography*, 50(3):305–324, 2009.
10. P. Gaudry. Fast genus 2 arithmetic based on Theta functions. *J. of Mathematical Cryptology*, 1:243–265, 2007.
11. Pierrick Gaudry and David Lubicz. The arithmetic of characteristic 2 Kummer surfaces and of elliptic Kummer lines. *Finite Fields Appl.*, 15(2):246–260, 2009.

12. R. Granger, F. Hess, R. Oyono, N. Thériault, and F. Vercauteren. Ate pairing on hyperelliptic curves. In *Advances in cryptology—EUROCRYPT 2007*, volume 4515 of *Lecture Notes in Comput. Sci.*, pages 430–447. Springer, Berlin, 2007.
13. Florian Hess. Pairing lattices. *Pairing-Based Cryptography–Pairing 2008*, pages 18–38, 2008.
14. Florian Hess, Nigel P. Smart, and Frederik Vercauteren. The eta pairing revisited. *IEEE Trans. Inform. Theory*, 52(10):4595–4602, 2006.
15. Jun-ichi Igusa. *Theta functions*. Springer-Verlag, New York, 1972. Die Grundlehren der mathematischen Wissenschaften, Band 194.
16. Shoji Koizumi. Theta relations and projective normality of Abelian varieties. *Amer. J. Math.*, 98(4):865–889, 1976.
17. Serge Lang. Reciprocity and correspondences. *Amer. J. Math.*, 80:431–440, 1958.
18. Tanja Lange. Formulae for arithmetic on genus 2 hyperelliptic curves. *Applicable Algebra in Engineering, Communication and Computing*, 15(5):295–328, 2005.
19. Stephen Lichtenbaum. Duality theorems for curves over p-adic fields. *Invent. Math.*, 7:120–136, 1969.
20. David Lubicz and Damien Robert. Efficient pairing computation with theta functions. *Algorithmic Number Theory*, 6197, 07 2010. 9th International Symposium, Nancy, France, ANTS-IX, July 19-23, 2010, Proceedings.
21. Victor S. Miller. The Weil pairing, and its efficient calculation. *J. Cryptology*, 17(4):235–261, 2004.
22. D. Mumford. On the equations defining abelian varieties. I. *Invent. Math.*, 1:287–354, 1966.
23. D. Mumford. On the equations defining abelian varieties. II. *Invent. Math.*, 3:75–135, 1967.
24. D. Mumford. *Abelian varieties*. Tata Institute of Fundamental Research Studies in Mathematics, No. 5. Published for the Tata Institute of Fundamental Research, Bombay, 1970.
25. David Mumford. *Tata lectures on theta I*, volume 28 of *Progress in Mathematics*. Birkhäuser Boston Inc., Boston, MA, 1983. With the assistance of C. Musili, M. Nori, E. Previato and M. Stillman.
26. David Mumford. *Tata lectures on theta II*, volume 43 of *Progress in Mathematics*. Birkhäuser Boston Inc., Boston, MA, 1984. Jacobian theta functions and differential equations, With the collaboration of C. Musili, M. Nori, E. Previato, M. Stillman and H. Umemura.
27. Joseph H. Silverman. *The arithmetic of elliptic curves*, volume 106 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1986. Corrected reprint of the 1986 original.
28. Frederik Vercauteren. Optimal pairings. *IEEE Trans. Inform. Theory*, 56(1):455–461, 2010.
29. Frederik Vercauteren. Optimal pairings. *Information Theory, IEEE Transactions on*, 56(1):455–461, 2010.
30. P. Wamelen. Equations for the Jacobian of a hyperelliptic curve. *AMS*, 350(8):3083–3106, 08 1999.