Chapter 5

Miller's algorithm for the Weil and Tate pairings

This chapter defines the Weil and Tate pairings and presents Miller's algorithm for computing them. As usual, we state the definitions in our context (on elliptic curves over finite fields), but the more general definitions are analogous (see [Sil09, Gal05]).

Notation. In this chapter we will use the notation $w_r(P, Q)$ for the (order r) Weil pairing of P and Q and $t_r(P, Q)$ for their (order r) Tate pairing, as this will help when discussing differences and relationships between them. After this chapter though, it will always be clear which pairing we mean and what the value of r is (the largest prime factor of $\#E(\mathbb{F}_q)$), so we will return to the notation most commonly seen in the literature and simply write $e(P, Q)$.

Both pairings make use of a special case of the following fact we recall from Chapter 3: a divisor $D = \sum_P n_P(P)$ is principal (i.e. the divisor of a function) if and only if $\sum_P n_P = 0$ and $\sum_P [n_P]P = \mathcal{O}$ on E. For any $m \in \mathbb{Z}$ and $P \in E$, it follows that there exists a function $f_{m,P}$ with divisor

$$(f_{m,P}) = m(P) - ([m]P) - (m-1)(\mathcal{O}), \tag{5.1}$$

where we note that for $m = 0$, one can take $f_{0,P} = 1$ with $(f_{0,P})$ the zero divisor.

Thus, if $P \in E[r]$, then $f_{r,P}$ has divisor

$$(f_{r,P}) = r(P) - r(\mathcal{O}). \tag{5.2}$$

Observe that $(f_{m+1,P}) - (f_{m,P}) = (P) + ([m]P) - ([m+1]P) - (\mathcal{O})$, which is exactly the divisor of the function $\ell_{[m]P,P}/v_{[m+1]P}$, where $\ell_{[m]P,P}$ and $v_{[m+1]P}$ are the sloped and vertical lines used in the chord-and-tangent addition of the point $[m]P$ and P (see Figure 5.1). This means we can build $f_{m+1,P}$ from $f_{m,P}$ via $f_{m+1,P} = f_{m,P} \cdot \frac{\ell_{[m]P,P}}{v_{[m+1]P}}$.

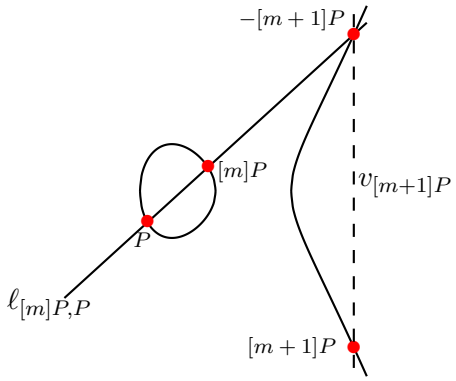


Figure 5.1: $\left(\frac{\ell_{[m]P,P}}{v_{[m+1]P}}\right) = (\ell_{[m]P,P}) - (v_{[m+1]P}) = (P) + ([m]P) - ([m+1]P) - (\mathcal{O})$.

Example 5.0.1 (Magma script). Let $q = 23$, and consider $E/\mathbb{F}_q : y^2 = x^3 + 17x + 6$ which has $\#E(\mathbb{F}_q) = 30$, and which has $P = (10, 7)$ as a point of order 5. Thus, we are guaranteed the existence of a function $f_{5,P}$ on E with divisor $(f_{5,P}) = 5(P) - 5(\mathcal{O})$. Starting with $m = 2$, we will build $f_{5,P}$ by using $f_{m+1,P} = f_{m,P} \cdot \frac{\ell_{[m]P,P}}{v_{[m+1]P}}$ (note that $(f_{1,P})$ is the zero divisor). The function $f_{2,P}$ with divisor $(f_{2,P}) = 2(P) - ([2]P) - (\mathcal{O})$ is the tangent line $l_{P,P}$ at P divided by the vertical line $v_{[2]P}$ through $[2]P$, which is $f_{2,P} = \frac{y+2x+19}{x+16}$. We compute the function $f_{3,P}$ as $f_{3,P} = f_{2,P} \cdot \frac{l_{P,[2]P}}{v_{[3]P}}$, where $l_{P,[2]P}$ is the chord through P and $[2]P$ and $v_{[3]P}$ is the vertical line at $[3]P$. Thus, $f_{3,P} = \frac{y+2x+19}{x+16} \cdot \frac{y+x+6}{x+16} = \frac{3y+x^2+9x+19}{x+16}$. Similarly, multiplication by the chord $l_{P,[3]P}$ through P and $[3]P$ and division by the vertical line $v_{[4]P}$ through $[4]P$ will advance us from $f_{3,P}$ to $f_{4,P}$, as $f_{4,P} = f_{3,P} \cdot \frac{l_{P,[3]P}}{v_{[4]P}} = \frac{3y+x^2+9x+19}{x+16} \cdot \frac{y+2x+19}{x+13} = \frac{(x+22)y+5x^2+3x+5}{x+13}$; this function has divisor $(f_{4,P}) = 4(P) - (4P) - 3(\mathcal{O})$. The last update we require is the function with divisor $(P) + (4P) - (5P) - (\mathcal{O})$, which would ordinarily be the quotient of lines in the addition of P and $4P$, but since P has order 5, we know that $P = -4P$, so this function actually has divisor $(P) + (-P) - 2(\mathcal{O})$. Thus, our last update

to the function is simply the vertical line at P, i.e. $(x-10)$, which gives the final function as $f_{5,P} = (x-10) \cdot \frac{(x+22)y+5x^2+3x+5}{x+13} = (x+22)y + 5x^2 + 3x + 5$; this function has a zero of order 5 on E at P, and a pole of order 5 on E at $\mathcal{O}$.

5.1 The Weil pairing

For a point $P \in E[r]$, the function $f_{r,P}$ with divisor $r(P) - r(\mathcal{O})$ is at the heart of both the Weil and Tate pairing definitions.

Definition 5.1 (The Weil pairing (over finite fields))**.** *Let $P, Q \in E(\mathbb{F}_{q^k})[r]$ and let D_P and D_Q be degree zero divisors with disjoint supports such that $D_P \sim (P) - (\mathcal{O})$ and $D_Q \sim (Q) - (\mathcal{O})$. There exist functions f and g such that $(f) = rD_P$ and $(g) = rD_Q$. The Weil pairing w_r is a map*

$$w_r : E(\mathbb{F}_{q^k})[r] \times E(\mathbb{F}_{q^k})[r] \to \mu_r,$$

defined as

$$w_r(P, Q) = \frac{f(D_Q)}{g(D_P)}.$$

Among other properties, the Weil pairing is bilinear and non-degenerate. We refer the reader to [Sil09, Ch. III, Prop. 8.1-8.2] for the proofs and full list of properties.

An important point to note is that we can not simply define the Weil pairing as $w_r(P, Q) = f_{r,P}(D_Q)/f_{r,Q}(D_P)$, because $(f_{r,P}) = r(P) - r(\mathcal{O})$ and $(f_{r,Q}) = r(Q) - r(\mathcal{O})$; this corresponds to the divisors $D_P = (P) - (\mathcal{O})$ and $D_Q = (Q) - (\mathcal{O})$, which does not adhere to the requirement that D_P and D_Q have disjoint supports.

Example 5.1.1 (Magma script). Let $q = 23$, and consider $E/\mathbb{F}_q : y^2 = x^3 - x$, which (is supersingular and therefore) has $\#E(\mathbb{F}_q) = q + 1 = 24$. The point $P = (2, 11)$ is a point of order $r = 3$ and the embedding degree with respect to r is $k = 2$. Take $\mathbb{F}_{q^2} = \mathbb{F}_q(i)$ with $i^2+1 = 0$, from which we obtain a point Q of order 3 (that is not in $\langle P \rangle$) as $Q = (21, 12i)$, which is actually in the trace zero subgroup, i.e. $\pi(Q) = [q]Q$. Suppose we wish to compute the Weil pairing $w_r(P, Q)$ of P and Q. For illustrative purposes, we will start by computing $f_{r,P}$ and $f_{r,Q}$ and then updating according to the above paragraph. Following the same technique as the last example, we get $f_{r,P}$ and $f_{r,Q}$ as $f_{r,P} = y+11x+13$ and $f_{r,Q} = y+11ix+10i$,

which have divisors $(f_{r,P}) = 3(P) - 3(\mathcal{O})$ and $(g_{r,P}) = 3(Q) - 3(\mathcal{O})$ respectively. We need to find divisors D_P and D_Q that have distinct supports but which are respectively equivalent to $(P) - (\mathcal{O})$ and $(Q) - (\mathcal{O})$. Note that only one of these divisors needs to be updated (so that its support does not contain $\mathcal{O}$), but we will update both in the name of symmetry. Thus, take two more (random) points in $E(\mathbb{F}_{q^2})$ as $R = (17i, 2i + 21)$ and $S = (10i + 18, 13i + 13)$, and set $D_P = (P + R) - (R)$ and $D_Q = (Q + S) - (S)$. We find f as a function with divisor D_P and g as a function with divisor D_Q as $f = f_{r,P}/(\ell_{P,R}/v_{P+R})^3$ and $g = g_{r,Q}/(\ell_{Q,S}/v_{Q+S})^3$ respectively, where $\ell_{P,R}/v_{P+R}$ is the quotient of the chord between P and R and the vertical line through $P + R$ (and similarly for $\ell_{Q,S}/v_{Q+S}$). We can now compute the Weil pairing according to Definition 5.1 as

$$\begin{aligned} w_r(P, Q) &= f(D_Q)/g(D_P) \\ &= \frac{f(Q + S) \cdot g(R)}{f(S) \cdot g(P + R)}. \\ &= 15i + 11. \end{aligned}$$

Observe that $(15i + 11)^3 = 1$ so $w_r(P, Q) \in \mu_r$. Repeating the whole process with $[2]P$ instead gives $w_r([2]P, Q) = 8i + 11 = w_r(P, Q)^2$, or with $[2]Q$ gives $w_r(P, [2]Q) = 8i + 11 = w_r(P, Q)^2$, or with both $[2]P$ and $[2]Q$ gives $w_r([2]P, [2]Q) = 15i + 11 = w_r(P, Q)^4 = w_r(P, Q)$, which is about as much of the bilinearity of w_r that we can illustrate in this toy example.

5.2 The Tate pairing

The formal definition of the Tate pairing requires that only one argument comes from the r-torsion. For our purposes, the other argument can be any point of $E(\mathbb{F}_{q^k})$, but we will soon see that in general it is still advantageous to choose both points from (distinct subgroups in) the r-torsion. In order to define the Tate pairing correctly though, we need to properly define the groups involved. We assume the standard setting that is of most interest to us: $k > 1$, $r \| \#E(\mathbb{F}_q)$ and, since there are r^2 points in the subgroup $E(\mathbb{F}_{q^k})[r]$, we usually have $r^2 \| \#E(\mathbb{F}_{q^k})$. Thus, let $h = \#E(\mathbb{F}_{q^k})/r^2$ be the cofactor that sends points in $E(\mathbb{F}_{q^k})$ to points

in $E(\mathbb{F}_{q^k})[r]$. Let $rE(\mathbb{F}_{q^k})$ be the *coset* of points in $E(\mathbb{F}_{q^k})$ defined by

$$rE(\mathbb{F}_{q^k}) = \{[r]P : P \in E(\mathbb{F}_{q^k})\}.$$

The number of elements in $rE(\mathbb{F}_{q^k})$ is h and it contains $\mathcal{O}$; from here we will simply denote this coset as rE. Following [Sco04], we can obtain another distinct coset of $E(\mathbb{F}_{q^k})$ by adding a random element R (not in $E[r]$) to each element of rE. In this way we can obtain precisely r^2 distinct, order h cosets. The quotient group E/rE is the group whose elements are these cosets. We note that elements belonging to each coset do not have the same order, nor do they form a (sub)group. In the quotient group E/rE, points belonging to the same coset (group element) can be used to *represent* the coset. Any two points in the same coset differ from one another by an element in rE, so one can think of E/rE as the set of equivalence classes of points in $E(\mathbb{F}_{q^k})$ under the equivalence relation $P_1 \equiv P_2$ if and only if $P_1 - P_2 \in rE$ [Gal05, IX.3].

Example 5.2.1 (Magma script). Let $q = 5$, and consider $E/\mathbb{F}_q : y^2 = x^3 - 3$, which has $\#E(\mathbb{F}_q) = 6$. Thus, taking $r = 3$ gives $k = 2$, so take $\mathbb{F}_{q^2} = \mathbb{F}_q(i)$, where $i^2 + 2 = 0$. Further, note that $\#E(\mathbb{F}_{q^2}) = 36 = hr^2$, so $h = 4$, and thus taking $rE = \{[3]P : P \in E(\mathbb{F}_{q^2})\}$ gives $rE = \{\mathcal{O}, (3i + 4, 0), (2i + 4, 0), (2, 0)\}$, with $\#rE = h$. Each of the other 8 cosets in E/rE are shown in Figure 5.2, where we importantly note that each coset has a unique *representative element* that lies in the r-torsion (see Figure 5.3). Consider the coset containing $P_1 = (2i, 4i + 3)$ $P_2 = (4, 1)$, $P_3 = (3, 2)$ and $P_4 = (3i, i + 3)\}$. All of the non-trivial pairwise differences are (defined by) $P_1 - P_2 = P_3 - P_4 = (3i + 4, 0)$, $P_1 - P_3 = P_2 - P_4 = (2i + 4, 0)$ and $P_1 - P_4 = P_2 - P_3 = (2, 0)$, which are all in rE.

For our purposes, $E[r]$ and the quotient group E/rE both have r^2 elements[1], but although it was the case in Example 5.2.1, it is not necessarily the case that the elements of $E[r]$ each represent a unique coset of E/rE (see [Gal05, IX.3] for a counterexample). However, if $r^2 \| \#E(\mathbb{F}_{q^k})$, then $E[r] \cap rE = \mathcal{O}$, which means that adding a unique torsion element to all of the elements in rE will generate a unique coset in E/rE. That is, $r^2 \| \#E(\mathbb{F}_{q^k})$ implies that $E[r]$ does represent E/rE (see [Gal05, Th. IX.22] for the proof in the supersingular scenario), and this will always be the case for us. This is particularly convenient when it comes to defining the Tate pairing, since the "second" group in the (order r) Tate

[1] In fact, they always have the same number of elements, but there are cases when the cardinality is not r^2 – see [Gal05, IX.3, IX.7.3]

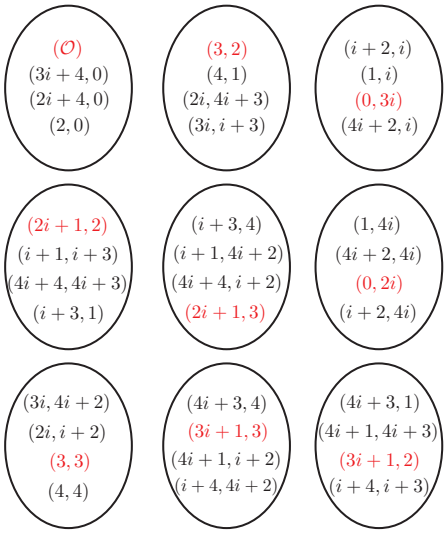


Figure 5.2: The r^2 cosets in the quotient group $E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k})$.

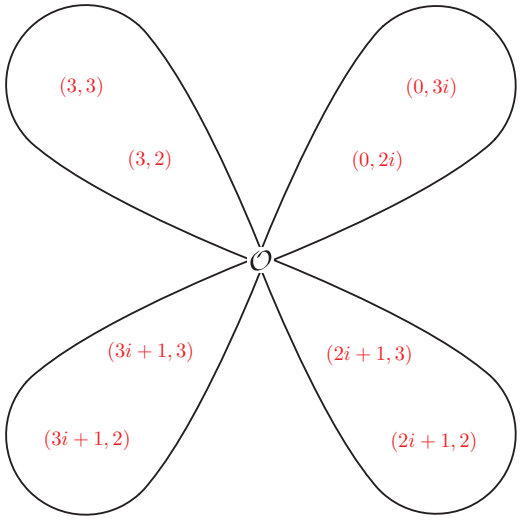


Figure 5.3: The r-torsion, where each $P \in E[r]$ is in a distinct coset of E/rE.

pairing is E/rE. As we will see after the definition, $E[r]$ representing E/rE allows us to take both groups from the r-torsion, which matches the somewhat simpler Weil pairing group definitions.

We note that although we refer to the following pairing as the Tate pairing throughout, it is often aptly called the Tate-Lichtenbaum pairing [Sil09, XI.9]. This is because Lichtenbaum [Lic69] specialised Tate's more general pairing to the case of Jacobians of curves (over local fields) which facilitates explicit computation [Gal05, IX.3].

Definition 5.2 (The Tate pairing (over finite fields))**.** *Let $P \in E(\mathbb{F}_{q^k})[r]$, from which it follows that there is a function f whose divisor is $(f) = r(P) - r(\mathcal{O})$. Let $Q \in E(\mathbb{F}_{q^k})$ be any representative in any equivalence class in $E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k})$, and let D_Q be a degree zero divisor defined over $\mathbb{F}_{q^k}$ that is equivalent to $(Q) - (\mathcal{O})$, but whose support is disjoint to that of (f). The Tate pairing t_r is a map*

$$t_r : E(\mathbb{F}_{q^k})[r] \times E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k}) \to \mathbb{F}_{q^k}^*/(\mathbb{F}_{q^k}^*)^r,$$

defined as

$$t_r(P, Q) = f(D_Q).$$

Again, we remark that among other properties, the Tate pairing is bilinear

and non-degenerate. We refer the reader to [Sil09, XI.9] and [Gal05, IX.4] for the proofs and full list of properties.

The quotient group $\mathbb{F}_{q^k}^*/(\mathbb{F}_{q^k}^*)^r$ is defined as we would expect. Namely, $(\mathbb{F}_{q^k}^*)^r$ is a subgroup of $\mathbb{F}_{q^k}^*$ defined as $(\mathbb{F}_{q^k}^*)^r = \{u^r : u \in \mathbb{F}_{q^k}^*\}$, so $\mathbb{F}_{q^k}^*/(\mathbb{F}_{q^k}^*)^r$ is the set of equivalence classes of $\mathbb{F}_{q^k}^*$ under the equivalence relation $a_1 \equiv a_2$ if and only if $a_1/a_2 \in (\mathbb{F}_{q^k}^*)^r$.

Example 5.2.2 (Magma script). We continue with the parameters from Example 5.2.1. Let $P = (3, 2) \in E[r]$ (see Figure 5.2) and let $Q = (i+1, 4i+2) \in E(\mathbb{F}_{q^k})$. The function $f : y + 2x + 2 = 0$ on E has divisor $3(P) - 3(\mathcal{O})$, so to compute the Tate pairing we need to find an appropriate $D_Q \sim (Q) - (\mathcal{O})$ but with $P, \mathcal{O} \notin \text{supp}(D_Q)$. Take R (randomly) as $R = (2i, i + 2)$, and let $D_Q = (Q + R) - (R)$, where $Q + R = (3i + 1, 2)$. The Tate pairing is computed as

$$t_r(P, Q) = f(D_Q) = \frac{f(Q + R)}{f(R)} = \frac{2 + 2 \cdot (3i + 1) + 2}{(i + 2) + 2 \cdot 2i + 2} = 4i + 4.$$

To illustrate bilinearity, computing $t_r(P, [2]Q)$ with $D_{[2]Q} = ([2]Q + R) - (R)$ where $[2]Q + R = (i + 2, i)$ gives

$$t_r(P, [2]Q) = f(D_{[2]Q}) = \frac{f([2]Q + R)}{f(R)} = \frac{i + 2 \cdot (i + 2) + 2}{(i + 2) + 2 \cdot 2i + 2} = 2i + 4,$$

or computing $t_r([2]P, Q)$, where $\tilde{f} = y + 3x + 3$ has divisor $\tilde{f} = r([2]P) - r(\mathcal{O})$, gives

$$t_r([2]P, Q) = \tilde{f}(D_Q) = \frac{\tilde{f}(Q + R)}{\tilde{f}(R)} = \frac{2 + 3 \cdot (3i + 1) + 3}{(i + 2) + 3 \cdot 2i + 3} = 3i + 2.$$

Note that $t_r(P, Q) = 4i + 4$, $t_r(P, [2]Q) = 2i + 4 = t_r(P, Q)^2$, but $t_r([2]P, Q) = 3i + 2$, i.e. $t_r(P, [2]Q), t_r([2]P, Q) \notin (\mathbb{F}_{q^k}^*)^r$, but $t_r(P, [2]Q)/t_r([2]P, Q) \in (\mathbb{F}_{q^k}^*)^r$, so $t_r(P, [2]Q) \equiv t_r([2]P, Q) \equiv t_r(P, Q)^2$ in $\mathbb{F}_{q^k}^*/(\mathbb{F}_{q^k}^*)^r$.

The above example illustrates an important point: in the context of cryptography, the standard Tate pairing has an undesirable property that its output lies in an equivalence class, rather than being a unique value. A necessary attribute for the Tate pairing to be useful in cryptography is that different parties must compute the exact same value under the bilinearity property, rather than values which are the same under the above notion of equivalence. Thus, to be suitable in practice, we must update the definition of the Tate pairing to make sure the

mapping produces unique values.

Definition 5.3 (The reduced Tate pairing)**.** *Let P, Q, f and D_Q be as in Definition 5.2. Over finite fields, the reduced Tate pairing T_r is a map*

$$T_r : E(\mathbb{F}_{q^k})[r] \times E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k}) \to \mu_r,$$

defined as

$$\begin{aligned} T_r(P,Q) &= t_r(P,Q)^{\#\mathbb{F}_{q^k}/r} \\ &= f_{r,P}(D_Q)^{(q^k-1)/r}. \end{aligned}$$

Exponentiating elements in $\mathbb{F}_{q^k}^*/(\mathbb{F}_{q^k}^*)^r$ to the power of $(q^k-1)/r$ kills r-th powers and sends the paired value to an exact r-th root of unity in μ_r.

From now on we will also take the second argument of the (reduced) Tate pairing from the r-torsion. In fact, we will further assume a Type 3 pairing. Therefore, in the pairing of P and Q, we will assume $P \in \mathbb{G}_1 = E[r] \cap \mathrm{Ker}(\pi - [1])$ and $Q \in \mathbb{G}_2 = E[r] \cap \mathrm{Ker}(\pi - [q])$. One should note that these choices are not restrictions, as far as what values the pairing can take: fixing P and letting Q run through $\langle Q \rangle$ (which has order r) will give each value in μ_r, and vice versa. Thus, for any $\tilde{P}, \tilde{Q}$ pair chosen from anywhere in the torsion, there exists a scalar $0 \leq a \leq r-1$ such that $T_r([a]P, Q) = T_r(P, [a]Q) = T_r(\tilde{P}, \tilde{Q})$.

Example 5.2.3 (Magma script). Let $q = 19$, $E/\mathbb{F}_q : y^2 = x^3 + 14x + 3$, giving $\#E(\mathbb{F}_q) = 20$, so take $r = 5$. The embedding degree is $k = 2$, so let $\mathbb{F}_{q^2} = \mathbb{F}_q(i)$ with $i^2 + 1 = 0$. The points $P = (17, 9)$ and $Q = (16, 16i)$ are in the r-torsion subgroups $\mathbb{G}_1 = E[r] \cap \mathrm{Ker}(\pi - [1])$ and $\mathbb{G}_2 = E[r] \cap \mathrm{Ker}(\pi - [q])$ respectively. The Tate pairing of P and Q is $t_r(P, Q) = 7i + 3$, whilst the reduced Tate pairing is $T_r(P, Q) = 15i + 2$. Let $\exp : \mathbb{F}_{q^k}^*/(\mathbb{F}_{q^k}^*)^r \to \mu_r$ be the map defined by the exponentiation $\exp : a \mapsto a^{(q^k-1)/r}$, i.e. $\exp : t_r(P, Q) \mapsto T_r(P, Q)$. Observe the difference between the Tate pairing t_r and reduced Tate pairing T_r for the following computations.

$$\begin{array}{cccc} t_r(P,Q)^4 & t_r([4]P,Q) & t_r(P,[4]Q) & t_r([2]P,[2]Q) \\ = 3i+7 & = 7i+16 & = 12i+3 & = 2i+14 \\ \Big\downarrow \exp & \Big\downarrow \exp & \Big\downarrow \exp & \Big\downarrow \exp \\ T_r(P,Q)^4 & T_r([4]P,Q) & T_r(P,[4]Q) & T_r([2]P,[2]Q) \\ = 4i+2 & = 4i+2 & = 4i+2 & = 4i+2 \end{array}$$

We note that none of the t_r lie in $(\mathbb{F}_{q^k})^5$, but the quotient of any two of them does lie there, so all the t_r pairings on the top level are equivalent in $\mathbb{F}_{q^k}^*/(\mathbb{F}_{q^k}^*)^r$. On the other hand, T_r ensures that each of the above pairings (that should be equivalent) take exactly the same value in $\mu_r \subset \mathbb{F}_{q^k}$.

From now on, when we say Tate pairing, we mean the reduced Tate pairing T_r in Definition 5.3.

5.3 Miller's algorithm

We briefly recap the pairing definitions from the previous two sections. For the r-torsion points P and Q, the Weil and Tate pairings are respectively computed as $\frac{f_{r,P}(D_Q)}{f_{r,Q}(D_P)}$ and $f_{r,P}(D_Q)^{(q^k-1)/r}$, where the divisors D_P and D_Q are chosen such that their supports are disjoint from the supports of $(f_{r,Q})$ and $(f_{r,P})$ respectively. For any points P and Q belonging to distinct subgroups in $E[r]$, we have already seen how to compute $f_{r,P}(D_Q)$ in the previous sections, but this was only for very small values of r. In practice r will be huge (i.e. at the very least 2^{160}), and since $f_{r,P}$ is a function of degree approximately r, it is not hard to see that computing this function as we did in the previous examples is impossible. In this section we describe Miller's algorithm [Mil04], which makes this computation very feasible. More precisely, the naive method of computing $f_{r,P}(D_Q)$ that we have been using has exponential complexity $O(r)$, whilst the algorithm we are about to describe for this computation has polynomial complexity $O(\log r)$. To put it simply, Miller's algorithm makes pairings practical; without this algorithm, secure cryptographic pairings would only be of theoretical value[2].

We start by referring back to the discussion at the beginning of this chapter. Following Equation (5.1), we saw that the divisor $(f_{m,P}) = m(P) - ([m]P) - (m-1)(\mathcal{O})$ could be updated to the divisor $(f_{m+1}) = (m+1)(P) - ([m+1]P) - m(\mathcal{O})$ by adding the divisor $(\ell_{[m]P,P}/v_{[m+1]P}) = (P) + ([m]P) - ([m+1]P) - (\mathcal{O})$; this corresponds to the multiplication of functions $f_{m+1} = f_m \cdot \ell_{[m]P,P}/v_{[m+1]P}$. Starting with $f_{2,P} = 2(P) - ([2]P) - (\mathcal{O})$ then, we can repeat this process roughly $r-1$ times to obtain the desired function $f_{r,P} = r(P) - ([r]P) - (r-1)(\mathcal{O}) = r(P) - r(\mathcal{O})$. We note that for the last step (i.e. when $m = r-1$) we have $f_{r-1,P} = (r-1)(P) - ([r-1]P) - (r-2)(\mathcal{O})$, so the required divisor is $(P) +$

[2]This is no longer entirely true. In 2007 Stange derived an alternative method to Miller's algorithm for efficiently computing the Tate pairing [Sta07], but it is currently less efficient than Miller's algorithm.

$([r-1]P) - 2(\mathcal{O})$ which corresponds to (a multiplication by!) the vertical line $v_{[r-1]P} = v_{-P} = v_P$; note that this is the same vertical line that appears on the denominator of $\ell_{[r-2]P,P}/v_{[r-1]P}$. Thus, the *pairing evaluation function* $f_{r,P}$ is the product

$$f_{r,P} = \ell_{[r-2]P,P} \cdot \prod_{i=1}^{r-3} \frac{\ell_{[i]P,P}}{v_{[i+1]P}}. \tag{5.3}$$

The first four sloped lines $\ell_{[i]P,P}$ and corresponding vertical lines $v_{[i+1]P}$ from the numerator and denominator of the product in (5.3) are shown in Figure 5.4 and Figure 5.5 respectively. We have seen that the product in (5.3) is (in the most

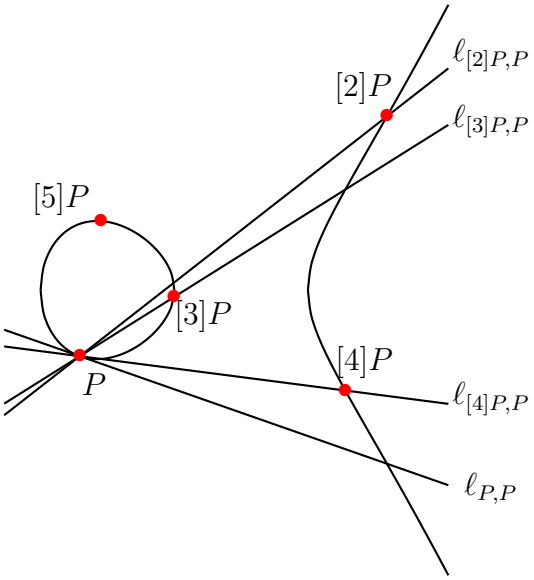


Figure 5.4: The first four sloped lines in the product (5.3).

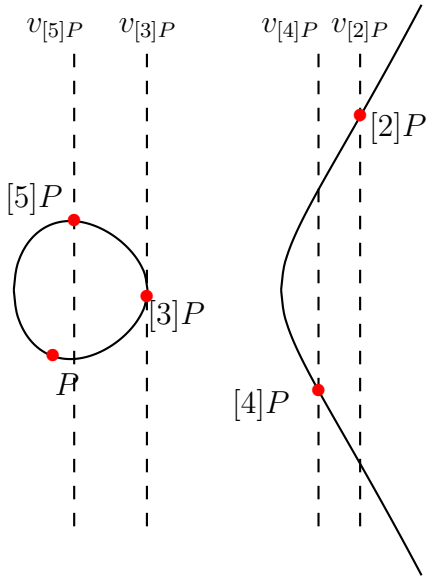


Figure 5.5: The first four vertical lines in the product (5.3).

naive way) built up incrementally by absorbing each of the $\frac{\ell_{[m]P,P}}{v_{[m+1]P}}$ terms into $f_{m,P}$ to increment to $f_{m+1,P}$, eventually arriving at $f_{r,P}$. Alternatively, it can help to see the divisor sum written out in full, to see the contributions of each

of the functions $\frac{\ell_{[i]P,P}}{v_{[i+1]P}}$ in the product all at once.

$$\begin{aligned}
\ell_{P,P}/v_{[2]P}: &\quad (P)+(P)-([2]P)-(\mathcal{O})\\
\ell_{[2]P,P}/v_{[3]P}: &\quad (P)+([2]P)-([3]P)-(\mathcal{O})\\
\ell_{[3]P,P}/v_{[4]P}: &\quad (P)+([3]P)-([4]P)-(\mathcal{O})\\
\vdots &\quad \vdots \quad \vdots \quad \vdots \quad \vdots\\
\ell_{[r-4]P,P}/v_{[r-3]P}: &\quad (P)+([r-4]P)-([r-3]P)-(\mathcal{O})\\
\ell_{[r-3]P,P}/v_{[r-2]P}: &\quad (P)+([r-3]P)-([r-2]P)-(\mathcal{O})\\
\ell_{[r-2]P,P} &\quad (P)+([r-2]P)+(-[r-1]P)-3(\mathcal{O})
\end{aligned}$$

When summing all of the above divisors, most of the inner terms cancel out with one another to leave $(r-1)(P)+(-[r-1]P)-r(\mathcal{O})$, and since $[r-1]P=-P$, we get the divisor of the product being $r(P)-r(\mathcal{O})$.

Roughly speaking, $f_{r,P}=g(x,y)/h(x,y)$, where g and h are degree r functions on E. The above method computes $f_{r,P}$ by successively increasing the degrees of g and h by one each time $f_{m,P}$ is incremented. This is why, when r is (exponentially) large, this naive method has exponential complexity. Miller's algorithm naturally overcomes this through the following observation. The function $f_{m,P}$ has m zeros at P and $(m-1)$ poles at $\mathcal{O}$. Rather than adding one zero and one pole via multiplying $f_{m,P}$ by linear functions, we can double the number of zeros at P and the number of poles at $\mathcal{O}$ if we instead square $f_{m,P}$. Observe that since $(f_{m,P})=m(P)-([m]P)-(m-1)(\mathcal{O})$, then

$$(f_{m,P}^2)=2m(P)-2([m]P)-2(m-1)(\mathcal{O}),$$

which is almost the same as $f_{2m,P}$, whose divisor is

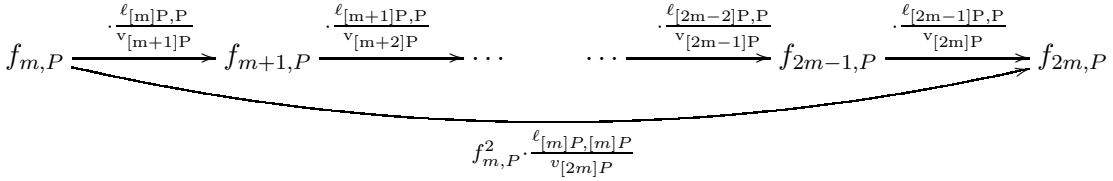
$$(f_{2m,P})=2m(P)-([2m]P)-(2m-1)(\mathcal{O});$$

the difference between the two divisors being $(f_{2m,P})-(f_{m,P}^2)=2([m]P)-([2m]P)-(\mathcal{O})$, which corresponds to a function with two zeros at $[m]P$, a pole at $[2m]P$ and another pole at $\mathcal{O}$. We have seen such a function many times already; this is simply the quotient of the tangent line at $[m]P$ and the vertical line at $[2m]P$ – the lines used to double the point $[m]P$. Thus, we can advance

from $f_{m,P}$ to $f_{2m,P}$ via

$$f_{2m,P} = f_{m,P}^2 \cdot \frac{\ell_{[m]P,[m]P}}{v_{[2m]P}}.$$

We depict the jump from $f_{m,P}$ to $f_{2m,P}$ (as opposed to the naive method of progressing one-by-one) below.



Since, for any m, we can now advance to either $f_{m+1,P}$ or $f_{2m,P}$ quickly, Miller observed that this gives rise to a double-and-add style algorithm to reach $f_{2,r}$ in $O(\log(r))$ steps. However, the degree of $f_{m,P}$ grows linearly in the size of m, so (en route to $m = r$) the function $f_{m,P}$ becomes too large to store explicitly. Thus, the last piece of the puzzle in Miller's derivation of the pairing algorithm was to, at every stage, evaluate $f_{m,P}$ at the given divisor, i.e. $f_{m,P}(D_Q)$. This means that at any intermediate stage of the algorithm we will not be storing an element of the function field $f_{m,P} \in \mathbb{F}_{q^k}(E)$, but rather its evaluation at D_Q which is the value $f_{m,P}(D_Q) \in \mathbb{F}_{q^k}$. At each stage then, the updates that build the function are evaluated at D_Q before being absorbed into intermediate pairing value that is carried through the routine. This is summarised in Algorithm 5.1 below, where the binary representation of r governs the double-and-add route taken to compute $f_{r,P}(D_Q)$, in an identical fashion to the standard double-and-add routine for scalar multiplications on E (see Example 2.1.8).

Miller's algorithm is essentially the straightforward double-and-add algorithm for elliptic curve point multiplication (see Example 2.1.8) combined with evaluations of the functions (the chord and tangent lines) used in the addition process.

Example 5.3.1 (Magma script). We will compute both the Weil and Tate pairings using Miller's algorithm. Let $q = 47$, $E/\mathbb{F}_q : y^2 = x^3 + 21x + 15$, which has $\#E(\mathbb{F}_q) = 51$, so we take $r = 17$. The embedding degree k with respect to r is $k = 4$, thus take $\mathbb{F}_{q^4} = \mathbb{F}_q(u)$ where $u^4 - 4u^2 + 5 = 0$. The point $P = (45, 23)$ has order 17 in $E(\mathbb{F}_q)$ which (because $k > 1$) means $P \in \mathbb{G}_1 = E[r] \cap \mathrm{Ker}(\pi - [1])$. The group order over the full extension field is $\#E(\mathbb{F}_{q^4}) = 3^3 \cdot 5^4 \cdot 17^2$, so take $h = 3^3 \cdot 5^4$ as the cofactor. Taking a random point from $E(\mathbb{F}_{q^4})$ and multiplying by h will (almost always) give a point $Q \in E[r]$, but it is likely to land outside of $\mathbb{G}_1 \cup \mathbb{G}_2$,

Algorithm 5.1 Miller's algorithm.

Input: $P \in E(\mathbb{F}_{q^k})[r]$, $D_Q \sim (Q) - (\mathcal{O})$ with support disjoint from $(f_{r,P})$, and $r = (r_{n-1} \dots r_1 r_0)_2$ with $r_{n-1} = 1$.
Output: $f_{r,P}(D_Q) \leftarrow f$.

1: $R \leftarrow P$, $f \leftarrow 1$.
2: **for** $i = n-2$ down to 0 **do**
3: Compute the line functions $\ell_{R,R}$ and $v_{[2]R}$ for doubling R.
4: $R \leftarrow [2]R$.
5: $f \leftarrow f^2 \cdot \frac{\ell_{R,R}}{v_{[2]R}}(D_Q)$.
6: **if** $r_i = 1$ **then**
7: Compute the line functions $\ell_{R,P}$ and v_{R+P} for adding R and P.
8: $R \leftarrow R + P$.
9: $f \leftarrow f \cdot \frac{\ell_{R,P}}{v_{R+P}}(D_Q)$.
10: **end if**
11: **end for**
12: **return** f.

so to move into $\mathbb{G}_2 = E[r] \cap \mathrm{Ker}(\pi - [q])$, we can use the anti-trace map (see Figure 4.4) and take $Q \leftarrow [k]Q - \mathrm{Tr}(Q)$. For example, $Q = (31u^2+29, 35u^3+11u)$ is one of 17 points in $\mathbb{G}_2$. The Tate pairing is $T_r(P,Q) = f_{r,P}(D_Q)^{(q^k-1)/r}$, whilst the Weil pairing is $w_r(P,Q) = \frac{f_{r,P}(D_Q)}{f_{r,Q}(D_P)}$. We will illustrate Miller's algorithm to compute $f_{r,P}(D_Q)$, since it appears in both. The binary representation of r is $r = (1,0,0,0,1)_2$. We will take D_Q as $D_Q = ([2]Q) - (Q)$, which clearly has support disjoint to $(f_{r,P})$ and is equivalent to $(Q) - (\mathcal{O})$. The table below shows the stages of Miller's algorithm for computing $f_{r,P}(D_Q)$: it shows the intermediate values of R, and of the function ℓ/v which corresponds to $\frac{\ell_{R,R}}{v_{[2]R}}$ or $\frac{\ell_{R,P}}{v_{R+P}}$ depending on whether we are in the doubling stage (steps 3-5 of Algorithm 5.1) or the addition stage (steps 6-10 of Algorithm 5.1); the table also shows the progression of the paired value f. To complete the Tate pairing, we compute

i/ r_i	steps of Alg. 5.1	point R	update ℓ/v	update at $[2]Q$ = $\frac{\text{update at } [2]Q}{\text{update at } Q}$ = $\frac{\ell(D_Q)}{v(D_Q)}$ = $\frac{\ell/v([2]Q)}{\ell/v(Q)}$	paired value f
	1	$(45,23)$			1
3/0	3-5	$(12,16)$	$\frac{y+33x+43}{x+35}$	$\frac{20u^3+21u^2+9u+4}{6u^3+19u^2+36u+33} = 41u^3+32u^2+2u+21$	$41u^3+32u^2+2u+21$
2/0	3-5	$(27,14)$	$\frac{y+2x+7}{x+20}$	$\frac{40u^3+18u^2+38u+9}{39u^3+8u^2+20u+18} = 4u^3+5u^2+28u+17$	$22u^3+27u^2+30u+33$
1/0	3-5	$(18,31)$	$\frac{y+42x+27}{x+29}$	$\frac{29u^3+15u^2+8u+14}{18u^3+32u^2+41u+30} = 6u^3+13u^2+33u+28$	$36u^3+2u^2+21u+37$
0/1	3-5	$(45,24)$	$\frac{y+9x+42}{x+2}$	$\frac{10u^3+3u^2+14u+19}{21u^3+26u^2+25u+20} = 46u^3+45u^2+u+20$	$10u^3+21u^2+40u+25$
	6-10	$\mathcal{O}$	$x+2$	$\frac{7u^2+27}{31u^2+31} = 6u^2+43$	$17u^3+6u^2+10u+22$
	12	$f_{r,P}(D_Q) \leftarrow 17u^3+6u^2+10u+22$			

$$t_r(P,Q) = f_{r,P}(D_Q)^{(q^k-1)/r} = (17u^3+6u^2+10u+22)^{287040} = 33u^3+43u^2+45u+39.$$

For the Weil pairing, we require another run of Miller's algorithm, this time reversing the roles of P and Q to compute $f_{r,Q}(D_P) = 2u^2 + 6u + 40$, which gives the Weil pairing as $w_r(P,Q) = \frac{f_{r,P}(D_Q)}{f_{r,Q}(D_P)} = \frac{17u^3+6u^2+10u+22}{2u^2+6u+40} = 22u^3+12u^2+32u+13$. Notice that, in line with Equation 5.3 (and the preceding discussion), the vertical line $x + 2 = 0$ that corresponds to the final addition in this example appears in the denominator of the previous ℓ/v function used for the doubling, and could therefore be cancelled out. We will see that this occurs in general, and is perhaps the least significant of many improvements to Miller's initial algorithm that have accelerated pairings over the last decade. Indeed, in Chapter 7 we will be looking at several more major optimisations to Miller's algorithm. 5.1.

5.4 Chapter summary

We started with the more simple description of the Weil pairing, before moving to the definition of the Tate pairing. This is because both the elliptic curve groups in the raw definition of the Weil pairing are torsion subgroups, which were discussed at length in the previous chapter. On the other hand, one of the groups in the general Tate pairing definition required us to introduce the quotient group $E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k})$. However, we soon showed that (for cases of cryptographic interest) it is no problem to represent this quotient group by a torsion subgroup, thereby unifying the group definitions needed for the Weil and Tate pairing and solidifying the choices of $\mathbb{G}_1 = E[r] \cap \mathrm{Ker}(\pi - [1])$ and $\mathbb{G}_2 = E[r] \cap \mathrm{Ker}(\pi - [q])$, which will be standard for the remainder of this text. We saw that at the heart of both the Weil and Tate pairings is the computation of the pairing evaluation function $f_{r,P}(D)$, where $P \in E$ and D is an appropriately defined divisor on E. We finished the chapter by presenting Miller's algorithm, which is the first practical algorithm to compute $f_{r,P}(D)$ for cases of cryptographic interest, and which remains the fastest algorithm for computing pairings to date.