Pairing for Icicle

Tomer Solberg

1 Introduction

Pairing functions are an important tool in elliptic curve cryptography. They have many applications, including but not limited to: BLS signatures [BLS01], ZK verification [Gro16], threshold encryption [CGPW24] and recently also witness encryption [LMP24]. Adding this functionality for Icicle can therefore be very advantageous for us, and open many avenues of usage for Icicle that have so far been closed off.

Of the curves currently supported by Icicle [Ing24], 4 are pairing friendly. These are

- BLS12-377 [HG21]
- BLS12-381 [Edg23]
- BN-254 [Wan24]
- BW6-761 [HG21]

2 General structure

Pairing functions are bilinear non-degenerate functions

$$e(\cdot,\cdot) : G_1 \times G_2 \to G_T \tag{2.1}$$

Where G_1, G_2, G_T are all cyclic groups of some prime order r. By bilinearity we mean that given 2 points $P_1, P_2 \in G_1$, and a point $Q \in G_2$ we have

$$e(P_1 + P_2, Q) = e(P1, Q) \oplus e(P_2, Q) \tag{2.2}$$

and similarly for $P \in G_1$ and $Q_1, Q_2 \in G_2$. Here $'+'$ is the group operation in G_1 which will generally be elliptic curve addition, while $'\oplus'$ is the group operation in G_T which will generally be finite field multiplication. By non-degenerate we mean that given two generators $g_i \in G_i,\ i = 1, 2$ we have that $e(g_1, g_2) \in G_T$ is also a generator.

2.1 EC pairing in Icicle

For the curves supported by Icicle, we take G_1 to be the scalar subgroup of order r of the elliptic curve over the base field $G_1 = E(\mathbb{F}_q)[r]$. G_2 is a scalar subgroup of what is known as a twist of the curve over a small (quadratic or cubic) extension field $G_2 = E'(\mathbb{F}_{q^\ell})[r]$. Lastly, G_T is the subgroup of order r roots of unity in the multiplicative subgroup of the k-th extension of the base field $G_T = \mu_r \subset \mathbb{F}^*_{q^k}$. k is known as the embedding degree, and

for the BLS and BN curves it takes the value $k = 12$, while for the BW curve we have $k = 6$.

The pairing we use for our curves is known as the optimized Ate pairing [MKHO07], and it is given by

$$e(P, Q) = f_{t-1,\bar{Q}}(P)^{(q^k-1)/r} \tag{2.3}$$

Where $t = q+1-\#E(\mathbb{F}_q)$ is the trace of Frobenius, $f_{t-1,\bar{Q}}(P) \in \mathbb{F}^*_{q^k}$ is calculated using (the optimized) Miller's algorithm. The point $\bar{Q} \in E(\mathbb{F}_{q^k})$ is the embedding of the point Q into $E(\mathbb{F}_{q^k})$. This embedding requires 'untwisting' the curve for $E'(\mathbb{F}_{q^\ell})$ to $E(\mathbb{F}_{q^k})$.

To summarize, we then need to define for each curve

1. The curve equation $E : y^2 + x^3 + ax + b$ and the parameters q, r, ℓ, k, t
2. The multiplication laws of the extension fields
3. The twisted curve equation $E' : y^2 + x^3 + a'x + b'$ and the untwisting map $\phi : Q \to \bar{Q}$

As well as Miller's algorithm which computes $f_{t-1,\bar{Q}}(P)$ and its optimizations for each curve. Note that for the implementation, E, E' are already implemented, as well as $\mathbb{F}_{q^\ell}$ so we only need to implement the higher field extensions, untwisting, Miller's loop, and some optimizations for the exponentiation by $(q^k - 1)/r$.

3 Extension field towers

To bulid the extension fields $\mathbb{F}_{q^\ell}, \mathbb{F}_{q^k}$ efficiently, we use a tower field approach, so that the base $T_0 = \mathbb{F}_q$ is a prime field, and every level of the tower is either a quadratic or cubic extension of the previous level $T_{i+1} = T_i[u_i]/(u_i^{\ell_i} - \beta_i)$ where $\ell_i \in \{2, 3\}$ and $\beta_i \in T_i$.

3.1 Quadratic extensions

A quadratic extension $T_{i+1} = T_i[u]/(u^2 - \beta)$ is one where $\ell = 2$, and it introduces a new element $u = \sqrt{\beta}$, so every element $x \in T_{i+1}$ can be written as $x = a + bu$ where $a, b \in T_i$. Alternatively we can write x as a tuple $x = (a, b)$. A common quadratic extension is the complex extension, where $\beta = -1$ and $u = i = \sqrt{-1}$. However it is important to note that in many cases we will have $\beta \neq -1$. The value β determines the multiplication law for the extension field

$$x \cdot y = (a + bu)(c + du) = ac + bdu^2 + (ad + bc)u \tag{3.1}$$
$$= (ac + \beta bd) + (ad + bc)u \tag{3.2}$$

where $(ac + \beta bd), (ad + bc) \in T_i$. Here we have 4 multiplications in the field T_i (not counting the multiplication by β which is generally a small constant), which can be reduced to 3 multiplications by a Karatsuba-like algorithm

$$x \cdot y = (ac + \beta bd) + ((a + b)(c + d) - ac - bd)u \tag{3.3}$$

3.2 Cubic extensions

A Cubic extension $T_{i+1} = T_i[w]/(w^3 - \beta)$ is one where $\ell = 3$, and it introduces two new elements w, w^2 where $w = \beta^{1/3}$, so every element $x \in T_{i+1}$ can be written as $x = a+bw+cw^2$ where $a, b, c \in T_i$. Alternatively we can write x as a tuple $x = (a, b, c)$. The value β determines the multiplication law for the extension field

$$x \cdot y = (a + bw + cw^2)(d + ew + fw^2) \tag{3.4}$$
$$= (ad + \beta(bf + ce)) + (ae + bd + \beta cf)w + (af + cd + be)w^2 \tag{3.5}$$
$$= A + Bw + Cw^2 \tag{3.6}$$

where $A, B, C \in T_i$. Here we have 9 multiplications in the field T_i (not counting the multiplications by β which is generally a small constant), which can be reduced to 6 multiplications by a Karatsuba-like algorithm [DhSD06]

$$A = ad + \beta((b + c)(e + f) - be - cf) \tag{3.7}$$
$$B = (a + b)(d + e) - ad - be + \beta cf \tag{3.8}$$
$$C = (a + c)(d + f) - ad - cf + be \tag{3.9}$$

In fact, a reduction to 5 multiplications is possible using Toom-Cook, but this is generally less efficient as it requires many more additions and multiplications by constants.

3.3 Cross-level arithmetic

Given an element $x \in T_i$ and an element in a lower level $y \in T_j$, $j < i$. It is possible to define arithmetic operations in the following recursive way. Write

$$x = \sum_{a=0}^{\ell_{i-1}} x_a u_{i-1}^a = (x_0, x_1, \ldots, x_{\ell_{i-1}}), \quad x_a \in T_{i-1} \tag{3.10}$$

and then

$$x + y = (x_0 + y) + \sum_{a=1}^{\ell_{i-1}} x_a u_{i-1}^a = (x_0 + y, x_1, \ldots, x_{\ell_{i-1}}) \tag{3.11}$$

$$x \cdot y = \sum_{a=0}^{\ell_{i-1}} x_a \cdot y \, u_{i-1}^a = (x_0 \cdot y, x_1 \cdot y, \ldots, x_{\ell_{i-1}} \cdot y) \tag{3.12}$$

Here the arithmetic is between elements of T_{i-1} and T_j. The recursion continues until we get arithmetic between elements of T_j.

3.4 Notation

Take for example the following tower: $T_0 = \mathbb{F}_q$, $T_1 = T_0[i]/(i^2 - \beta_1)$, $T_2 = T_1[v]/(v^3 - \beta_2)$, $T_3 = T_2[u]/(u^2 - \beta_3)$. We will generally represent $x \in T_3$ as a multivariate polynomial

of T_0 elements as follows

$$x = x_0 + x_1 u \qquad x_0, x_1 \in T_2 \tag{3.13}$$
$$= (x_{00} + x_{01} v + x_{02} v^2) + (x_{10} + x_{11} v + x_{12} v^2) u \qquad x_{ij} \in T_1 \tag{3.14}$$
$$= ((x_{000} + x_{001} i) + (x_{010} + x_{011} i) v + (x_{020} + x_{021} i) v^2) + \tag{3.15}$$
$$((x_{100} + x_{101} i) + (x_{110} + x_{111} i) v + (x_{120} + x_{121} i) v^2) u \qquad x_{ijk} \in T_0 \tag{3.16}$$

Or alternatively as a sum $x = \sum_{j=0}^{11} x_j u_j$ where $x_j \in T_0$ and u_j is the monomial basis

$$u_j = (1, i, v, iv, v^2, iv^2, u, iu, vu, ivu, v^2 u, iv^2 u) \tag{3.17}$$

4 BLS12-381

The BLS12-381 curve is given by the equation $E : y^2 = x^3 + 4$ over the base field $\mathbb{F}_q$ with

$$z = -\text{0xd201000000010000}, \qquad t - 1 = z \tag{4.1}$$
$$q = \frac{1}{3}(z-1)^2 (z^4 - z^2 + 1) + z \tag{4.2}$$
$$= \text{0x1a0111ea397fe69a4b1ba7b6434bacd764774b84f38512bf6730d2a0f6b0f6241...} \tag{4.3}$$
$$\text{eabfffeb153ffffb9feffffffffaaab} \tag{4.4}$$
$$r = z^4 - z^2 + 1 = \text{0x73eda753299d7d483339d80809a1d80553bda402fffe5bfeffffffff00000001} \tag{4.5}$$
$$\ell = 2, \quad k = 12 \tag{4.6}$$

The structure of extension field towers is given by

$$T_1 = \mathbb{F}_{q^2} = \mathbb{F}_q[i]/(i^2 + 1) \tag{4.7}$$
$$T_2 = \mathbb{F}_{q^6} = T_1[v]/(v^3 - (i + 1)) \tag{4.8}$$
$$T_3 = \mathbb{F}_{q^{12}} = T_2[u]/(u^2 - v) \tag{4.9}$$

The twisted curve equation is $E' : y^2 = x^3 + 4(1 + i)$, where $x, y \in T_1 = \mathbb{F}_{q^2}$. For $Q = (x, y) \in E'$ we have $\bar{Q} = (X, Y) = (x/u^2, y/u^3)$. This can be seen by plugging $x = Xu^2,\ y = Yu^3,\ u^6 = 1 + i$ into E' to see that (X, Y) indeed satisfy $E : Y^2 = X^3 + 4$. To get a more explicit expression for $\bar{Q}$, note that

$$\frac{x}{u^2} = \frac{x}{v} = \frac{xv^2}{1+i} \tag{4.10}$$
$$= \frac{x(1-i)}{2} v^2 + 0u \in T_3 \tag{4.11}$$
$$\frac{y}{u^3} = \frac{yu^3}{1+i} \tag{4.12}$$
$$= 0 + \frac{y(1-i)}{2} vu \in T_3 \tag{4.13}$$

Note that $\frac{x(1-i)}{2}, \frac{y(1-i)}{2} \in T_1$, and that by multiplying them by v^2, v we get elements of T_2.

5 BLS12-377

The BLS12-377 curve is given by the equation $E : y^2 = x^3 + 1$ over the base field $\mathbb{F}_q$ with

$$z = \text{0x8508c00000000001}, \qquad t - 1 = z \tag{5.1}$$

$$q = \frac{1}{3}(z-1)^2(z^4 - z^2 + 1) + z \tag{5.2}$$

$$= \text{0x01ae3a4617c510eac63b05c06ca1493b1a22d9f300f5138f1ef3622fba0948...} \tag{5.3}$$

$$\text{00170b5d44300000008508c00000000001} \tag{5.4}$$

$$r = z^4 - z^2 + 1 = \text{0x12ab655e9a2ca55660b44d1e5c37b00159aa76fed00000010a11800000000001} \tag{5.5}$$

$$\ell = 2, \quad k = 12 \tag{5.6}$$

The structure of extension field towers is given by (Note that here $i = \sqrt{-5}$)

$$T_1 = \mathbb{F}_{q^2} = \mathbb{F}_q[i]/(i^2 + 5) \tag{5.7}$$

$$T_2 = \mathbb{F}_{q^6} = T_1[v]/(v^3 - i) \tag{5.8}$$

$$T_3 = \mathbb{F}_{q^{12}} = T_2[u]/(u^2 - v) \tag{5.9}$$

The twisted curve equation is $E' : y^2 = x^3 + 1/i$, where $x, y \in T_1 = \mathbb{F}_{q^2}$. For $Q = (x, y) \in E'$ we have $\bar{Q} = (X, Y) = (xu^2, yu^3)$. This can be seen by plugging $x = X/u^2$, $y = Y/u^3$, $u^6 = i$ into E' to see that (X, Y) indeed satisfy $E : Y^2 = X^3 + 1$. To get a more explicit expression for $\bar{Q}$, note that

$$xu^2 = xv + 0u \in T_3 \tag{5.10}$$

$$yu^3 = 0 + yvu \in T_3 \tag{5.11}$$

Note that $x, y \in T_1$, and that by multiplying them by v we get elements of T_2.

6 BN-254

The BN-254 curve is given by the equation $E : y^2 = x^3 + 3$ over the base field $\mathbb{F}_q$ with

$$z = 4965661367192848881, \qquad t - 1 = 6z^2 \tag{6.1}$$

$$q = 36z^4 + 36z^3 + 24z^2 + 6z + 1 \tag{6.2}$$

$$= 21888242871839275222246405745257275088696311157297823662689037894645226208583 \tag{6.3}$$

$$r = 36z^4 + 36z^3 + 18z^2 + 6z + 1 \tag{6.4}$$

$$= 21888242871839275222246405745257275088548364400416034343698204186575808495617 \tag{6.5}$$

$$\ell = 2, \quad k = 12 \tag{6.6}$$

The structure of extension field towers is given by

$$T_1 = \mathbb{F}_{q^2} = \mathbb{F}_q[i]/(i^2+1) \tag{6.7}$$
$$T_2 = \mathbb{F}_{q^6} = T_1[v]/(v^3-(i+9)) \tag{6.8}$$
$$T_3 = \mathbb{F}_{q^{12}} = T_2[u]/(u^2-v) \tag{6.9}$$

The twisted curve equation is $E' : y^2 = x^3 + 3/(9+i)$, where $x, y \in T_1 = \mathbb{F}_{q^2}$. For $Q = (x, y) \in E'$ we have $\bar{Q} = (X, Y) = (xu^2, yu^3)$. This can be seen by plugging $x = X/u^2,\ y = Y/u^3,\ u^6 = 9+i$ into E' to see that (X, Y) indeed satisfy $E : Y^2 = X^3 + 3$. To get a more explicit expression for $\bar{Q}$, note that

$$xu^2 = xv + 0u \in T_3 \tag{6.10}$$
$$yu^3 = 0 + yvu \in T_3 \tag{6.11}$$

Note that $x, y \in T_1$, and that by multiplying them by v we get elements of T_2.

7 BW6-761

The BW6-761 curve is given by the equation $E : y^2 = x^3 - 1$ over the base field $\mathbb{F}_q$ with

$$z = \text{0x8508c00000000001} \tag{7.1}$$
$$r = \frac{1}{3}(z-1)^2(z^4 - z^2 + 1) + z \tag{7.2}$$
$$= \text{0x01ae3a4617c510eac63b05c06ca1493b1a22d9f300f5138f1ef3622fba0948...} \tag{7.3}$$
$$\text{00170b5d44300000008508c00000000001} \tag{7.4}$$
$$t = z^5 - 3z^4 + 3z^3 - z + 3 \tag{7.5}$$
$$q = ((t+13r)^2 + 3(t/3+9r)^2)/4 \tag{7.6}$$
$$= \text{0x122e824fb83ce0ad187c94004faff3eb926186a81d14688528275ef8087be41707...} \tag{7.7}$$
$$\text{ba638e584e91903cebaff25b423048689c8ed12f9fd9071dcd3dc73ebff2e98a116...} \tag{7.8}$$
$$\text{c25667a8f8160cf8aeeaf0a437e6913e6870000082f49d00000000008b} \tag{7.9}$$
$$\ell = 3, \quad k = 6 \tag{7.10}$$

The structure of extension field towers is given by

$$T_1 = \mathbb{F}_{q^3} = \mathbb{F}_q[v]/(v^3+4) \tag{7.11}$$
$$T_2 = \mathbb{F}_{q^6} = T_1[u]/(u^2-v) \tag{7.12}$$

The twisted curve equation is $E' : y^2 = x^3 + 4$, where $x, y \in T_1 = \mathbb{F}_{q^2}$. For $Q = (x, y) \in E'$ we have $\bar{Q} = (X, Y) = (x/u^2, y/u^3)$. This can be seen by plugging $x = Xu^2,\ y = Yu^3,\ u^6 = -4$ into E' to see that (X, Y) indeed satisfy $E : Y^2 = X^3 - 1$. To get a more explicit expression for $\bar{Q}$, note that

$$\frac{x}{u^2} = \frac{x}{v} = \frac{-xv^2}{4} + 0u \in T_2 \tag{7.13}$$

$$\frac{y}{u^3} = 0 + \frac{-yv}{4}u \in T_2 \tag{7.14}$$

Note that $\frac{-xv^2}{4}, \frac{-yv}{4} \in T_1$.

8 Miller's algorithm

Miller's algorithm takes two points $P, Q \in E$ and an integer s as input, as well as the parameters a, b of E. s is given as a binary expansion $s = \sum_{j=0}^{t} b_j 2^t$ with $b_t = 1$. The algorithm returns the value $f_{s,P}(Q) \in \mathbb{F}_{q^k}^*$ which is then exponentiated to get the pairing result. The algorithm is shown in algorithm 1 (all operations here are field operations in $\mathbb{F}_{q^k}$) [Aut24]

Algorithm 1 Calculating $f_{s,P}(Q)$, for subgroup of size r of the curve $y^2 = x^3 + ax + b$

function MillerAlg($P = (x_P, y_P), Q = (x_Q, y_Q)$)
  if $P = O$ or $Q = O$ or $P = Q$ **then**
    return $f_{s,P}(Q) \leftarrow (-1)^s$
  end if
  $(x_T, y_T, f_1, f_2) \leftarrow (x_P, y_P, 1, 1)$
  for $j \leftarrow t-1, \ldots, 0$ **do**
    $(x_{2T}, y_{2T}, m) \leftarrow 2(x_T, y_T)$ ▷ Point doubling in E, m is the slope
    $f_1 \leftarrow f_1^2(y_Q - y_T - m(x_Q - x_T))$
    $f_2 \leftarrow f_2^2(x_Q + 2x_T - m^2)$ ▷ $\mathbb{F}_{q^k}$ arithmetic
    $(x_T, y_T) \leftarrow (x_{2T}, y_{2T})$
    if $b_j == 1$ **then**
      $(x_{T+P}, y_{T+P}, m) \leftarrow (x_T, y_T) + (x_P, y_P)$ ▷ Point addition in E (slope m)
      $f_1 \leftarrow f_1(y_Q - y_T - m(x_Q - x_T))$
      $f_2 \leftarrow f_2(x_Q + x_P + x_T - m^2)$ ▷ $\mathbb{F}_{q^k}$ arithmetic
      $(x_T, y_T) \leftarrow (x_{T+P}, y_{T+P})$
    end if
  end for
  $f_1 \leftarrow f_1(x_Q - x_T)$ ▷ $\mathbb{F}_{q^k}$ arithmetic
  return $f_{s,P}(Q) \leftarrow \frac{f_1}{f_2}$
end function

However, the computation of pairing we can optimize this algorithm. Recall that for the last step of the pairing calculation, we exponentiate the result by $(q^k - 1)/r$. It turns out that the denominator in Miller's algorithm f_2 will always satisfy $f_2^{(q^k-1)/r} = 1$. The last multiplication $f_1 \leftarrow f_1(x_Q - x_T)$ can also be skipped, as it is claimed in §6.5 of [LD07] that $(x_Q - x_T)^{\frac{q^k-1}{r}} = 1$. Also, since q is an odd prime we have $(-1)^{q^k-1} = 1$. We can therefor optimize our pairing by calculating $\hat{f}_{s,P}(Q)$ which is only the numerator in Miller's algorithm. This is given in algorithm 2 [LD07], where we also change the notation to use

the line function $\ell_{T,P}(Q) = y_Q - y_T - m_{T,P}(x_Q - x_T)$ where $m_{T,P}$ is the slope of the line which arises from the addition $T + P$.

Algorithm 2 Calculating $\hat{f}_{s,P}(Q)$, for subgroup of size r of the curve $y^2 = x^3 + ax + b$

function OptMillerAlg($P = (x_P, y_P), \bar{Q} = (x_Q, y_Q)$)
  if $P = O$ or $Q = O$ or $P = Q$ **then**
    return $\hat{f}_{s,P}(Q) \leftarrow 1$
  end if
  $(x_T, y_T, f_1) \leftarrow (x_P, y_P, 1)$
  for $j \leftarrow t - 1, \ldots, 0$ **do**
    $(x_{2T}, y_{2T}) \leftarrow 2(x_T, y_T)$ ▷ Point doubling in E
    $f_1 \leftarrow f_1^2 \cdot \ell_{T,T}(Q)$
    $(x_T, y_T) \leftarrow (x_{2T}, y_{2T})$
    if $b_j == 1$ **then**
      $(x_{T+P}, y_{T+P}) \leftarrow (x_T, y_T) + (x_P, y_P)$ ▷ Point addition in E
      $f_1 \leftarrow f_1 \cdot \ell_{T,P}(Q)$
      $(x_T, y_T) \leftarrow (x_{T+P}, y_{T+P})$
    end if
  end for
  return $\hat{f}_{s,P}(Q) \leftarrow f_1$
end function

For the Tate pairing, we would need to calculate $f_{r,P}(\bar{Q})$. Here, Miller's algorithm undergoes one main optimization, where all point operations (additions/doublings) in the loop are done in $E(\mathbb{F}_q)$, and following field arithmetic operations are done using cross-level arithmetic, with only $x_Q, y_Q, f_1 \in \mathbb{F}_{q^k}$, and the other elements are in $\mathbb{F}_q$.

However, the curves we use are all constructed for particularly optimized calculation of the Ate pairing, where we calculate $f_{t-1,\bar{Q}}(P)$. Here, point operations will be in $\mathbb{F}_{q^k}$, however there will be quite fewer of them, as the loop runs over the bits of $t - 1$ rather than r. $t - 1$ for these curves is much smaller than r, and has much fewer '1'-valued bits, as well as other specific optimizations for these particular curves. Note that we will also require the line functions $\ell_{T,Q}(P), \ell_{T,T}(P)$

8.1 Miller loop for BLS12 curves

For BLS curves an optimization due to [Sco19], which suggests to perform point operations in projective coordinates. This allows for point operations to be in $G_2 = E'(\mathbb{F}_{q^2})$, with the untwisting operation included in the calculation of the line functions. This also utilizes sparsity in the extension tower construction of the result. For BLS12-377, given $(X_T, Y_T, Z_T), (X_Q, Y_Q, Z_Q)$ are the projective coordinates of $T, Q \in E'(\mathbb{F}_{q^2})$, and $P = (x_P, y_P) \in E(\mathbb{F}_q)$ is given in affine coordinates. The line functions will be (left for

addition, right for doubling)

$$\ell_{T,Q}(P) = A + Bu + Cvu \qquad \ell_{T,T}(P) = A + Bu + Cvu \tag{8.1}$$
$$A = (X_T - Z_T X_Q)y_P \qquad A = -2Y_T Z_T y_P \tag{8.2}$$
$$B = -(Y_T - Z_T Y_Q)x_P \qquad B = 3X_T^2 x_P \tag{8.3}$$
$$C = (Y_T - Z_T Y_Q)X_Q - (X_T - Z_T X_Q)Y_Q \qquad C = 3b' Z_T^2 - Y_T^2 \tag{8.4}$$

where b' is the b value of E', or $1/i$ for BLS12-377. Due to the sparsity of the line function, which contains only 3 of the 6 possible $\mathbb{F}_{q^2}$ terms in $\mathbb{F}_{q^{12}}$, it is advantageous whenever $b_j = 1$ to first calculate $\ell_{T,T}(P)\ell_{T,Q}(P)$ before multiplying the result by f_1^2.

For BLS12-381, we get the same values for A, B, C, except now $b' = 4(1+i)$. The difference is in the tower field structure of the line function, which now becomes (both for addition and doubling)

$$\ell_{T,Q}(P) = A + Bu^{-1} + Cu^{-3} = A - \frac{B}{4}uv^2 - \frac{C}{4}uv \tag{8.5}$$

8.2 Miller loop for BN254

An optimization for BN254 [Wan24] gives

$$m(P,Q) = f_{6z+2,Q}(P) \cdot \ell_{[6z+2]Q,\pi_q(Q)}(P) \cdot \ell_{[6z+2]Q+\pi_q(Q),-\pi_q^2(Q)}(P) \tag{8.6}$$

where $\pi_q(Q) = (x_Q^q, y_Q^q)$ is the Frobenius map of the point Q. So that the Miller loop needs to follow the following steps

1. Calculate $f_{6z+2,Q}(P)$ using Algorithm 2, while storing $T \leftarrow [6z+2]Q$
2. Multiply the result by $\ell_{T,\pi_q(Q)}(P)$, while storing $T \leftarrow T + \pi_q(Q)$
3. Multiply the result by $\ell_{T,-\pi_q^2(Q)}(P)$

This is followed as usual by the final exponentiation to get $e(P,Q) = m(P,Q)^{(q^k-1)/r}$.

8.3 Miller loop for BW6-761

An optimization for BW6-761 [Hou22] gives

$$m(P,Q) = f_{z+1+(t-1)(z^3-z^2-z),Q}(P) \tag{8.7}$$

This is fastest in 2-NAF representation, which is the unique representation $s = \sum_i s_i 2^i$ where $s_i \in \{-1, 0, 1\}$ and no two adjacent s_i's are non-zero. For this representation, we use algorithm 3 as a variant of the Miller algorithm.

9 Optimizing the final exponentiation

The last exponentiation can be optimized in the following way [LD07]. First, note that the polynomial $x^k - 1$ can be decomposed into cyclotomic polynomials in the following way

Algorithm 3 Calculating $\hat{f}_{s,P}(Q)$, for subgroup of size r of the curve $y^2 = x^3 + ax + b$

function OptMillerAlg2NAF($P = (x_P, y_P), \bar{Q} = (x_Q, y_Q)$)
  if $P = O$ or $Q = O$ or $P = Q$ **then**
    return $\hat{f}_{s,P}(Q) \leftarrow 1$
  end if
  $(x_T, y_T, f_1) \leftarrow (x_P, y_P, 1)$
  for $j \leftarrow t-1, \ldots, 0$ **do**
    $(x_{2T}, y_{2T}) \leftarrow 2(x_T, y_T)$ ▷ Point doubling in E
    $f_1 \leftarrow f_1^2 \cdot \ell_{T,T}(Q)$
    $(x_T, y_T) \leftarrow (x_{2T}, y_{2T})$
    if $b_j \neq 1$ **then**
      $(x_{T+b_jP}, y_{T+b_jP}) \leftarrow (x_T, y_T) + (x_P, b_j y_P)$ ▷ Point addition in E
      $f_1 \leftarrow f_1 \cdot \ell_{T,b_jP}(Q)$
      $(x_T, y_T) \leftarrow (x_{T+b_jP}, y_{T+b_jP})$
    end if
  end for
  return $\hat{f}_{s,P}(Q) \leftarrow f_1$
end function

$$x^k - 1 = \prod_{d|k} \Phi_d(x) \tag{9.1}$$

For example for $k = 12$ we get the following cyclotomic decomposition

$$\Phi_1(x) = x - 1 \qquad \Phi_2(x) = x + 1 \qquad \Phi_3(x) = x^2 + x + 1 \tag{9.2}$$

$$\Phi_4(x) = x^2 + 1 \qquad \Phi_6(x) = x^2 - x + 1 \qquad \Phi_{12}(x) = x^4 - x^2 + 1 \tag{9.3}$$

$$x^{12} - 1 = \Phi_1(x)\Phi_2(x)\Phi_3(x)\Phi_4(x)\Phi_6(x)\Phi_{12}(x) \tag{9.4}$$

Next, note that the embedding degree k is chosen as the smallest extension field degree for which there is a subgroup of order r. This means that r is a divisor of $\Phi_k(q)$ and not of any $\Phi_d(q)$ with $d < k$. We can then write

$$\frac{q^k - 1}{r} = \frac{\Phi_k(q)}{r} \cdot \prod_{d|k, d<k} \Phi_d(q) \tag{9.5}$$

Which for $k = 6, 12$ gives

$$\frac{q^6 - 1}{r} = \frac{q^2 - q + 1}{r} \cdot (q^4 + q^3 - q - 1) \tag{9.6}$$

$$\frac{q^{12} - 1}{r} = \frac{q^4 - q^2 + 1}{r} \cdot (q^8 + q^6 - q^2 - 1) \tag{9.7}$$

Next, note that $\frac{\Phi_k(q)}{r} \cdot \prod_{d|k, d<k} \Phi_d(q)$ is a polynomial in q. This means that we can use a theorem in finite fields which states that for $a, b \in \mathbb{F}_{q^n}$ for any n, we have $(a+b)^q = a^q + b^q$.

This means that given $x \in \mathbb{F}_{q^k}$, we can take a decomposition $x = \sum_{j=0}^{k-1} x_j u_j$ where $x_j \in \mathbb{F}_q$ and u_j is the monomial basis of the expansion field. We then use Fermat's little theorem that says that for any $a \in \mathbb{F}_q$ we have $a^q = a$ to finally write

$$x^q = \sum_{j=0}^{k-1} x_j^q u_j^q = \sum_{j=0}^{k-1} x_j u_j^q \tag{9.8}$$

$$x^{q^4+q^3-q-1} = \frac{(\sum_{j=0}^{k-1} x_j u_j^{q^4})(\sum_{j=0}^{k-1} x_j u_j^{q^3})}{(\sum_{j=0}^{k-1} x_j u_j^q)x} \tag{9.9}$$

$$x^{q^8+q^6-q^2-1} = \frac{(\sum_{j=0}^{k-1} x_j u_j^{q^8})(\sum_{j=0}^{k-1} x_j u_j^{q^6})}{(\sum_{j=0}^{k-1} x_j u_j^{q^2})x} \tag{9.10}$$

We can the precompute the powers of the basis elements and store them in a table, so that this exponentiation can be done with a small number of operations ($3k$ base field multiplications, 3 extension field multiplication, and one extension field inversion).

Lastly, we exponentiate by $\Phi_k(q)/r$. This can also be somewhat optimized, by using the structure of q, r specific to the curve. For the BLS curves we have $q = yr + z$ where $y = (z-1)^2/3$ and $r = z^4 - z^2 + 1$. This gives

$$\frac{q^4 - q^2 + 1}{r} = yq^3 + yzq^2 + y(z^2-1)q + yz(z^2-1) + 1 \tag{9.11}$$

We can now use the same trick as before to calculate the q powers which gives

$$x^{\frac{q^4-q^2+1}{r}} = \left(\sum_{j=0}^{k-1} x_j u_j^{q^3}\right)^y \left(\sum_{j=0}^{k-1} x_j u_j^{q^2}\right)^{yz} \left(\sum_{j=0}^{k-1} x_j u_j^q\right)^{y(z^2-1)} x^{yz(z^2-1)+1} \tag{9.12}$$

This in general is not a major improvement, as we have replaced one exponentiation of size $\sim q^4$ by 4 exponentiations of size $\sim q$. However in the case of the BLS curves, z, y have very few 1-valued bits in their binary representations, so these exponentiations can in fact be accelerated.

9.1 Amortizing the exponentiation

Another optimization we can do for the final exponentiation is to amortize it for several calculations of pairing. The idea is that usually pairing calculations come as a test that a relation of the following form holds

$$e(P, Q) = e(R, S) \tag{9.13}$$

In this case we can rewrite it as

$$e(P, Q)e(-R, S) = 1 \tag{9.14}$$

Now, if we separate the Miller calculation and the exponentiation, so we write in short $e(P, Q) = f(P, Q)^\alpha$, we can see that it is enough to check

$$(f(P,Q)f(-R,S))^{\alpha} = 1 \tag{9.15}$$

So it is enough to calculate the exponent once. More generally, for a relation of the form

$$\prod_i e(P_i, Q_i) = \prod_j e(R_j, S_j) \tag{9.16}$$

it is enough to check

$$\left(\prod_i f(P_i, Q_i) \prod_j f(-R_j, S_j)\right)^{\alpha} = 1 \tag{9.17}$$

So it is enough to exponentiate once per such relation, regardless of how many pairings are included. This means, that other than a pairing function, we would like to provide the user with a "test pairing relation" function, which amortizes the exponent.

10 Testing and future work

To test the implementation, we need to check two things:

1. Given generators $g_1 \in G_1,\ g_2 \in G_2$, we must check that $g_T = e(g_1, g_2)$ is an r-th root of unity in $\mathbb{F}_{q^k}$, that is $g_T^r = 1$ and $g_T^j \neq 1$ for all $j < r$ (it's enough to check for values of j which divide r).

2. Given arbitrary integers m, n, we must check that $e(mg_1, ng_2) = g_T^{mn}$.

Alternatively, the results can also be tested by comparing to the Arkworks implementation [Ark21] which contains all of these curves.

Lastly, we would like to benchmark our implementation and compare it to others in terms of speed. Benchmarking of all of our curves using different implementations was done by Gnark [Hou21].

For future work, we intend to integrate Miller loop optimizations for multi-pairings, i.e. products of pairing functions, together with parallelization optimizations.

11 Some theory

I present here some theory, mostly due to [Cos], in a non-formal and non-rigorous way, mostly for intuition. This is not required for the implementation.

11.1 Counting EC points

Intuitively we expect $\#E(\mathbb{F}_q) \approx q$, since there are q possible value of x, and for about half of them $x^3 + ax + b$ will be a quadratic residue in $\mathbb{F}_q$. For each such QR, we will get two points on the curve $(x, \pm\sqrt{x^3 + ax + b})$.

To get a more precise estimation, let's introduce a few definitions

- The closure of a finite field $\mathbb{F}_q$ is the field containing all roots of polynomials over $\mathbb{F}_q$. This is
$$\bar{\mathbb{F}}_q = \bigcup_{i\in\mathbb{N}} \mathbb{F}_{q^i} \tag{11.1}$$
- The closure of a curve is $E = E(\bar{\mathbb{F}}_q)$
- Frobenius Endomorphism $\pi : E \to E$ such that $\pi(x, y) = (x^q, y^q)$

Note that from Fermat's little theorem π act trivially on $E(\mathbb{F}_q)$, and in fact it acts non trivially on $E \setminus E(\mathbb{F}_q)$. Similarly π^i acts non-trivially on $E \setminus \bigcup_{j\le i} E(\mathbb{F}_{q^j})$. So we get

$$P \in E(\mathbb{F}_q) \Leftrightarrow \pi(P) = P \Leftrightarrow (1-\pi)P = 0 \tag{11.2}$$
$$\Rightarrow \#E(\mathbb{F}_q) = \#ker(1-\pi) = \deg(1-\pi) \tag{11.3}$$

Where the last equality comes from separability. We thus have a version of CS inequality

$$|\deg(1-\pi) - \deg(1) - \deg(\pi)| \le 2\sqrt{\deg(1)\deg(\pi)} \tag{11.4}$$
$$|\#E(\mathbb{F}_q) - (1+q)| \le 2\sqrt{q} \tag{11.5}$$

The value $t = q + 1 - \#E(\mathbb{F}_q)$ is called the trace of Frobenius, and there's a theorem that says we can find a curve E for any integer value of t in that range.

One can also calculate the characteristic polynomial of π which is

$$\pi^2 - t\pi + q = 0 \tag{11.6}$$
$$(x^{q^2}, y^{q^2}) - t(x^q, y^q) + q(x, y) = 0 \tag{11.7}$$

Now to calculate $\#E(\mathbb{F}_q)$ we use Schoof's algorithm. The rough idea is to take this equation modulo some prime p_ℓ to get $t_\ell = t \mod p_\ell$, do this for enough primes and when $\prod_\ell p_\ell \ge 4\sqrt{q}$ we have enough data to use CRT to determine t. There's some more complexity to it but this is the rough idea.

Knowing $\#E(\mathbb{F}_q)$ allows us to also know $\#E(\mathbb{F}_{q^i})$ for any i using a theorem due to Weil which states that the sequence given by

$$t_0 = 2, \quad t_1 = q + 1 - \#E(\mathbb{F}_q) \tag{11.8}$$
$$t_i = t_1 t_{i-1} - q t_{i-2} \tag{11.9}$$

gives $\#E(\mathbb{F}_{q^i}) = q^i + 1 - t_i$.

11.2 r-torsion

So now we know the size $\#E(\mathbb{F}_q)$, and say it contain some prime factor r, meaning that $E(\mathbb{F}_q)$ contains a cyclic subgroup of order r. Can we find more points of order r in $E(\bar{\mathbb{F}}_q)$? The set of all of these is known as the r-torsion:

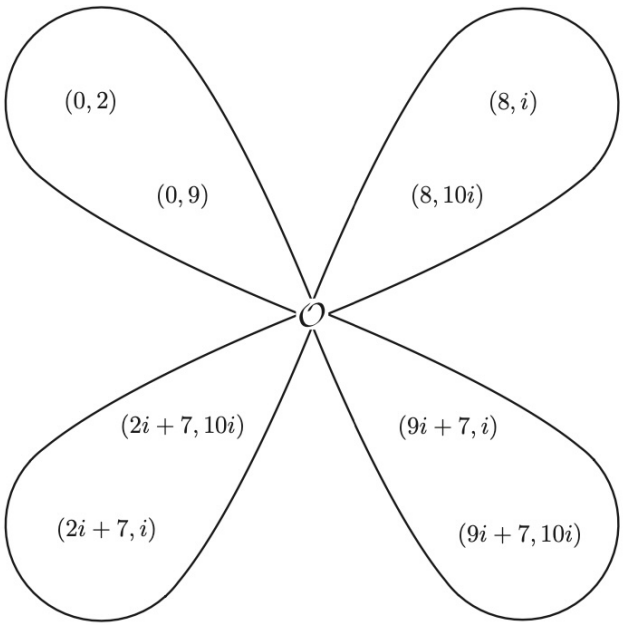
$$E[r] = \{P \in E : rP = 0\} \tag{11.10}$$

The way to find those is as follows

1. Find the smallest k such that $r|(q^k - 1)$
2. This gives us the smallest extension field $\mathbb{F}_{q^k}$ which contains all r-th roots of unity in $\bar{\mathbb{F}}_q$: $\mu_r \subset \mathbb{F}_{q^k}$
3. It turns out that $E[r] \subset E(\mathbb{F}_{q^k})$

In fact, there's a theorem that tells us exactly the size and structure of $E[r] \cong \mathbb{Z}_r \times \mathbb{Z}_r$. This means that given 2 elements $u, v \in E[r]$ that are not in the same cyclic group, we can build all of $E[r]$ as $\alpha u + \beta v$ with $\alpha, \beta \in \mathbb{Z}_r$. This tells us that $\#E[r] = r^2$. Since O is in every subgroup, we have $r^2 - 1 = (r+1)(r-1)$ non-O points. This tells us that we have $r+1$ subgroups of order r (each contain $r-1$ non-O points).

Let's illustrate this with an example. Consider $q = 11$ and $E : y^2 = x^3 + 4$. We get $\#E(\mathbb{F}_q) = 12$. Take $r = 3$. Order-3 points in $E(\mathbb{F}_q)$ are $\{O, (0,2), (0,8)\}$. Since $3 \nmid (11-1)$ we get $k \neq 1$. Since $3|(11^2 - 1)$ we get $k = 2$. Define $\mathbb{F}_{q^2} = \mathbb{F}_q[i]/(i^2+1)$. The subgroups are



This flower structure is a general structure of the r-torsion. The idea of pairing comes from connecting two leaves of this diagram, which is done by utilizing different homomorphisms between them.

11.3 Types of pairing

We can look at the solutions to equation (11.6) modulo r, which are simply the solutions restricted to $E[r]$. Since $r|\#E(\mathbb{F}_q)$ we get that

$$\pi^2 - t\pi + q = \pi^2 - (q+1)\pi + q = (\pi - 1)(\pi - q) \mod r \tag{11.11}$$

We can thus identify the base field subgroup of $E[r]$ as $G_1 = E[r] \cap \ker(\pi - 1)$. We can further identify another special subgroup with the eigenvalue q, $G_q = E[r] \cap \ker(\pi - q)$.

The different types of pairing are

1. Type 1: This is when $G_2 = G_1$. It is simpler, but it can only be calculated efficiently when there's a simple mapping from G_1 to another leaf. This happens for supersingular curves, which are curves with $t = 0$ or $\#E(\mathbb{F}_q) = q + 1$. For these there exists a simple homomorphism. Example: $q = 59$, $E : y^2 = x^3 + x$, we have $\phi(x, y) = (-x, iy)$ such that $\phi(G_1) = G_2$ and we can calculate $e : G_1 \times G_1 \to G_T$ using $\hat{e} : G_1 \times G_2 \to G_T$.

2. Type 2: This is pairing where we take G_2 to be any other subgroup but the two special ones we identified $G_2 \neq G_q, G_1$

3. Type 3: This is pairing with $G_2 = G_q$. It is orders of magnitude more efficient than other types of pairing and this is what we generally use.

4. Type 4: Take $G_2 = E[r]$ (the entire r-torsion)

11.4 How pairing is calculated

To explain this we first need to introduce the notion of divisors. A divisor on E is a formal sum $D = \sum_{P \in E} n_P(P)$ with only finitely many non-zero $n_P \in \mathbb{Z}$. The set of divisors $Div(E)$ is a group with integer addition and $D = 0$ as the identity. We define $\deg(D) = \sum_P n_P$ and the support $Supp(D) = \{P \in E : n_P \neq 0\}$. Given a function f on E define $ord_P(f)$ as the multiplicity of f at P. This is positive if $f(P) = 0$, negative if f has a pole at P, and zero otherwise. We define the divisor of f

$$(f) = \sum_P ord_P(f)(P) \tag{11.12}$$

Examples

Look at the line $\ell : y = \lambda x + \nu$. This line intersects the curve at 3 points $P, Q - (P + Q)$ which are its zeros. It also has a triple pole at infinity (we get this from projective coordinates), so we get

$$(\ell) = (P) + (Q) + (-P - Q) - 3(O) \tag{11.13}$$

a vertical line $v : x = \alpha$ intersects the curve at two points $P, -P$, and has a double pole at infinity, so

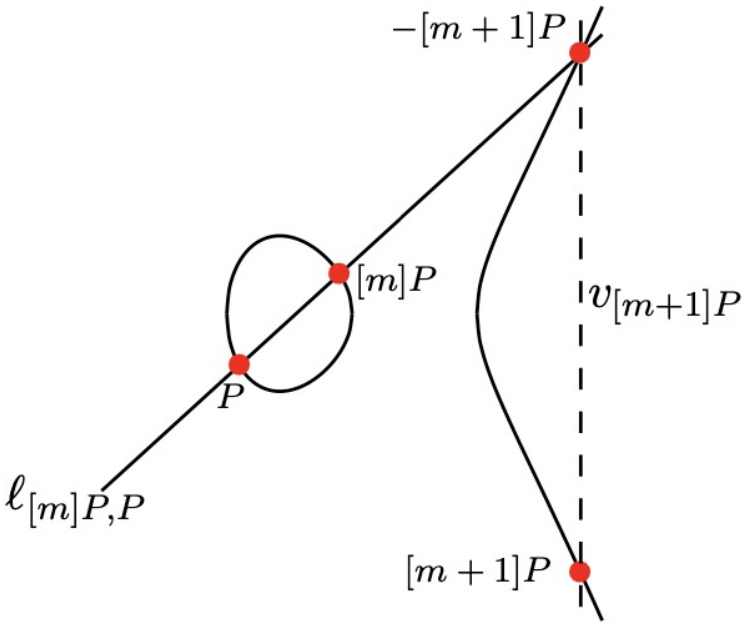
$$(v) = (P) + (-P) - 2(O) \tag{11.14}$$

In general, for every function f we have $\deg(f) = 0$ (however not every divisor of degree 0 comes from a function). Algebra between functions translates to divisor algebra

$$(fg) = (f) + (g) \tag{11.15}$$

$$(f) = 0 \Leftrightarrow f = const \tag{11.16}$$

For a divisor $D = \sum_P n_P(P)$ with $\deg(D) = 0$ to correspond to a function, it must obey an additional condition, that $\sum_P n_P P = O$ (This is an EC point sum). We get that (f)



which obeys this condition determines a function f up to a non-zero scalar multiplication. Now, the calculation of pairing come from a theorem that says there exists a function $f_{m,P}$ with divisor

$$(f_{m,P}) = m(P) - (mP) - (m-1)(O) \tag{11.17}$$

Where we define $f_{0,P} = 1$. If we take $P \in E[r]$ we get for this function $(f_{r,P}) = r(P) - r(O)$, which clearly obeys the condition to be the divisor of a function, since $rP = O$. Now since

$$\left(\frac{f_{m+1,P}}{f_{m,P}}\right) = (P) + (mP) - ((m+1)P) - (O) \tag{11.18}$$

$$\left(\frac{\ell_{P,mP}}{v_{(m+1)P}}\right) = (P) + (mP) - ((m+1)P) - (O) \tag{11.19}$$

We get

$$f_{m+1,P} = f_{m,P}\frac{\ell_{P,mP}}{v_{(m+1)P}} \tag{11.20}$$

This is the base step for Miller's algorithm, which allows us to calculate $f_{r,P}$.

11.5 The Weil, Tate and Ate pairing

We define the Weil, Tate and Ate pairings

$$e_W(P,Q) = (-1)^r \frac{f_{r,P}(Q)}{f_{r,Q}(P)} \tag{11.21}$$

$$e_T(P,Q) = f_{r,P}(Q)^{\frac{q^k-1}{r}} \tag{11.22}$$

$$e_A(P,Q) = f_{t-1,Q}(P)^{\frac{q^k-1}{r}} \tag{11.23}$$

where in the Ate pairing, t denotes the trace of Frobenius. The optimization we use for the Tate and Ate pairings relies on the fact that for vertical lines $(v_{(m+1)P})^{\frac{q^k-1}{r}} = 1$, so we don't need to calculate denominators.

References

[Ark21] Arkworks. arkworks::algebra, 2021. https://github.com/arkworks-rs/algebra.

[Aut24] Least Authority. The moonmath manual to zk-snarks, 2024. https://github.com/LeastAuthority/moonmath-manual/releases/latest/download/main-moonmath.pdf.

[BLS01] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the weil pairing. In *Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13, 2001, Proceedings*, volume 2248 of *Lecture Notes in Computer Science*, pages 514–532. Springer, 2001. https://iacr.org/archive/asiacrypt2001/22480516.pdf.

[CGPW24] Arka Rai Choudhuri, Sanjam Garg, Guru-Vamsi Policharla, and Mingyuan Wang. Practical mempool privacy via one-time setup batched threshold encryption. Cryptology ePrint Archive, Paper 2024/1516, 2024. https://eprint.iacr.org/2024/1516.

[Cos] Craig Costello. Pairings for beginners. https://static1.squarespace.com/static/5fdbb09f31d71c1227082339/t/5ff394720493bd28278889c6/1609798774687/PairingsForBeginners.pdf.

[DhSD06] Augusto Jun Devegili, Colm Ó hÉigeartaigh, Michael Scott, and Ricardo Dahab. Multiplication and squaring on pairing-friendly fields. Cryptology ePrint Archive, Paper 2006/471, 2006. https://eprint.iacr.org/2006/471.

[Edg23] Ben Edgington. Bls12-381 for the rest of us, 2023. https://hackmd.io/@benjaminion/bls12-381.

[Gro16] Jens Groth. On the size of pairing-based non-interactive arguments. Cryptology ePrint Archive, Paper 2016/260, 2016. https://eprint.iacr.org/2016/260.

[HG21] Youssef El Housni and Aurore Guillevic. Families of SNARK-friendly 2-chains of elliptic curves. Cryptology ePrint Archive, Paper 2021/1359, 2021. https://eprint.iacr.org/2021/1359.

[Hou21] Youssef El Housni. Benchmarking pairing-friendly elliptic curves libraries, 2021. https://hackmd.io/@gnark/eccbench.

[Hou22] Youssef El Housni. What changes for bw6-761?, 2022. https://hackmd.io/@gnark/BW6-761-changes.

[Ing24] Ingonyama. Icicle curves parameters, 2024. https://github.com/ingonyama-zk/icicle/tree/main/icicle/include/icicle/curves/params.

[LD07] B. Lynn and Stanford University. Computer Science Department. *On the Implementation of Pairing-based Cryptosystems*. Stanford University, 2007. https://crypto.stanford.edu/pbc/thesis.pdf.

[LMP24] Yanyi Liu, Noam Mazor, and Rafael Pass. On witness encryption and laconic zero-knowledge arguments. Cryptology ePrint Archive, Paper 2024/1932, 2024. https://eprint.iacr.org/2024/1932.

[MKHO07] Seiichi Matsuda, Naoki Kanayama, Florian Hess, and Eiji Okamoto. Optimised versions of the ate and twisted ate pairings. Cryptology ePrint Archive, Paper 2007/013, 2007. https://eprint.iacr.org/2007/013.

[Sco19] Michael Scott. Pairing implementation revisited. Cryptology ePrint Archive, Paper 2019/077, 2019.

[Wan24] Jonathan Wang. Bn254 for the rest of us, 2024. https://hackmd.io/@jpw/bn254.