A Variant of Miller's Formula and Algorithm

John Boxall, Nadia El Mrabet, Fabien Laguillaumie, Duc-Phong Le

A Variant of Miller's Formula and Algorithm

John Boxall[1], Nadia El Mrabet[2⋆], Fabien Laguillaumie[3], and Duc-Phong Le[4⋆]

[1] LMNO – Université de Caen Basse-Normandie, France
john.boxall@unicaen.fr
[2] LIASD – Université Paris 8, France
nelmrabe@mime.univ-paris8.fr
[3] GREYC – Université de Caen Basse-Normandie, France
fabien.laguillaumie@unicaen.fr
[4] Faculty of Information Technology
College of Technology,
Vietnam National University, Vietnam
phongld@vnu.edu.vn

Abstract. Miller's algorithm is at the heart of all pairing-based cryptosystems since it is used in the computation of pairing such as that of Weil or Tate and their variants. Most of the optimizations of this algorithm involve elliptic curves of particular forms, or curves with even embedding degree, or having an equation of a special form. Other improvements involve a reduction of the number of iterations.
In this article, we propose a variant of Miller's formula which gives rise to a *generically* faster algorithm for any pairing friendly curve. Concretely, it provides an improvement in cases little studied until now, in particular when denominator elimination is not available. It allows for instance the use of elliptic curve with embedding degree not of the form $2^i 3^j$, and is suitable for the computation of *optimal pairings*. We also present a version with denominator elimination for even embedding degree. In our implementations, our variant saves between 10% and 40% in running time in comparison with the usual version of Miller's algorithm without any optimization.

1 Introduction

Pairings were first introduced into cryptography in Joux' seminal paper describing a tripartite (bilinear) Diffie-Hellman key exchange [20]. Since then, the use of cryptosystems based on bilinear maps has had a huge success with some notable breakthroughs such as the first identity-based encryption scheme [10]. Nevertheless, pairing-based cryptography has a reputation of being inefficient, because it is computationally more expensive than cryptography based on modular arithmetic. On the other hand, the use of pairings seems to be essential in the definition of protocols with specific security properties and also allows one to reduce bandwidth in certain protocols.

⋆ This work was done while these two authors held post-doc positions at the Université de Caen Basse-Normandie.

Ever since it was first described, Miller's algorithm [24] has been the central ingredient in the calculation of pairings on elliptic curves. Many papers are devoted to improvements in its efficiency. For example, it can run faster when the elliptic curves are chosen to belong to specific families (see for example [4, 6, 12]), or different coordinate systems (see for example [18, 13, 7]). Another standard method of improving the algorithm is to reduce the number of iterations by introducing pairings of special type, for example particular *optimal pairings* [26, 17, 16] or using addition chains (see for example [8]).

In this paper we study a variant of Miller's algorithm for elliptic curves which is *generically* faster than the usual version. Instead of using the formula $f_{s+t} = f_s f_t \frac{\ell_{s,t}}{v_{s+t}}$ (see Subsection 2.2 for notation and Lemma 1) on which the usual Miller algorithm is based, our variant is inspired by the formula $f_{s+t} = \frac{1}{f_{-s} f_{-t} \ell_{-s,-t}}$ (see Lemma 2 for a proof). An important feature is that the only vertical line that appears is f_{-1}, in other words the vertical line passing through P, and even this does not appear explicitly except at initialization. We shall see in § 3.1 why it does not appear in the addition step. Our algorithm is of particular interest to compute the Ate-style pairings [3, 17] on elliptic curves with small embedding degrees k, and in situations where denominator elimination using a twist is not possible (for example on curves with embedding degree prime to 6). A typical example is the case of *optimal pairings* [26], which by definition only require about $\log_2(r)/\varphi(k)$ (where r is the group order) iterations of the basic loop. If k is prime, then $\varphi(k \pm 1) \leq \frac{k+1}{2}$ which is roughly $\frac{\varphi(k)}{2} = \frac{k-1}{2}$, so that at least twice as many iterations are necessary if curves with embedding degrees $k \pm 1$ are used instead of curves of embedding degree k.

The paper is organized as follows. In Section 2 we recall some background on pairings, and recall the usual Miller algorithm (Figure 1). In Section 3 we explain and analyze generically our version of Miller's algorithm, which is resumed by the pseudocode in Figure 2 when the elliptic curve is given in Jacobian coordinates. Section 4 discusses a variant without denominators applicable when k is even (see Figure 5). Section 5 describes some numerical experiments and running times; in an example with $k = 18$ and r having 192 bits, the algorithm of Figure 5 is roughly 40% faster than the usual Miller algorithm, and about as fast as the algorithm of [4].

Further work is needed to see whether many of the recent ideas used to improve the usual Miller algorithm can be adapted to the variant presented here. We believe that doing so should lead to further optimizations.

2 Background on Pairings

2.1 Basics on Pairings

We briefly recall the basic definitions and some examples of pairings used in cryptography. For further information, see for example [9, 11].

We let $r \geq 2$ denote an integer which, unless otherwise stated, is supposed to be prime. We let $(\mathbb{G}_1, +)$, $(\mathbb{G}_2, +)$ and $(\mathbb{G}_T, \cdot)$ denote three finite abelian groups,

which are supposed to be of order r unless otherwise indicated. A *pairing* is a map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ such that $e(P_1 + P_2, Q) = e(P_1, Q)e(P_2, Q)$ and $e(P, Q_1 + Q_2) = e(P, Q_1)e(P, Q_2)$ for all P, P_1, $P_2 \in \mathbb{G}_1$ and for all Q, Q_1, $Q_2 \in \mathbb{G}_2$. We say that the pairing e is *left non degenerate* if, given $P \in \mathbb{G}_1$ with $P \neq 1$, there exists $Q \in \mathbb{G}_2$ with $e(P, Q) \neq 1$. The notion of a right degenerate pairing is defined similarly and e is said to be *non degenerate* if it is both left and right non degenerate

We recall briefly one of the most frequent choices for the groups $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ in pairing-based cryptography. Here, $\mathbb{G}_1$ is the group generated by a point P of order r on an elliptic curve E defined over a finite field $\mathbb{F}_q$ of characteristic different to r. Thus, $\mathbb{G}_1 \subseteq E(\mathbb{F}_q)$ is cyclic of order r but, in general, the whole group $E[r]$ of points of order dividing r of E is not rational over $E(\mathbb{F}_q)$. Recall that the *embedding degree* of E (with respect to r) is the smallest integer $k \geq 1$ such that r divides $q^k - 1$. A result of Balasubramanian and Koblitz [2] asserts that, when $k > 1$, all the points of $E[r]$ are rational over the extension $\mathbb{F}_{q^k}$ of degree k of $\mathbb{F}_q$. The group $\mathbb{G}_2$ is chosen as another subgroup of $E[r]$ of order r. Finally, $\mathbb{G}_T$ is the subgroup of order r in $\mathbb{F}_{q^k}^\times$; it exists and is unique, since r divides $q^k - 1$ and $\mathbb{F}_{q^k}^\times$ is a cyclic group.

Let $P \in E(\mathbb{F}_q)$ be an r-torsion point, let D_P be a degree zero divisor with $D_P \sim [P] - [O_E]$, and let f_{r,D_P} be such that $\operatorname{div} f_{r,D_P} = rD_P$. Let Q be a point of $E(\mathbb{F}_{q^k})$ (not necessarily r-torsion) and $D_Q \sim [Q] - [O_E]$ of support disjoint with D_P. Consider

$$e_r^T(P, Q) = f_{r,D_P}(D_Q). \tag{1}$$

Weil reciprocity shows that if D_Q is replaced by $D'_Q = D_Q + \operatorname{div} h \sim D_Q$, then (1) is multiplied by $h(D_P)^r$. So the value is only defined up to r-th powers. Replacing D_P by $D'_P = D_P + \operatorname{div} h$ changes f_{r,D_P} to $f_{r,D'_P} = f_{r,D_P}h^r$, and the value is well-defined modulo multiplication by r-th powers. If then Q is replaced by $Q + rR$, the value changes again by an r-th power. This leads to adapting the range and domain of e_r^T as follows.

Theorem 1. *The Tate pairing is a map*

$$e_r^T : E(\mathbb{F}_q)[r] \times E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k}) \to \mathbb{F}_{q^k}^\times/(\mathbb{F}_{q^k}^\times)^r$$

satisfying the following properties:

1. *Bilinearity,*
2. *Non-degeneracy,*
3. *Compatibility with isogenies.*

The *reduced* Tate pairing computes the unique rth root of unity belonging to the class of $f_{r,D_P}(D_Q)$ modulo $(\mathbb{F}_{q^k}^\times)^r$ as $f_{r,D_P}(D_Q)^{(q^k-1)/r}$. In practice, we take Q to lie in some subgroup $\mathbb{G}_2$ of order r of $E(\mathbb{F}_{q^k})$ that injects into $E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k})$ *via* the canonical map. The more popular Ate pairing [3] and its variants (see [23] for instance) are optimized versions of the Tate pairing when restricted to Frobenius eigenspaces. Besides its use in cryptographic protocols,

the Tate pairing is also useful in other applications, such as walking on isogeny volcanoes [19], which can be used in the computation of endomorphism rings of elliptic curves.

However, in this article we concentrate on the computation of $f_{n,D_P}(D_Q)$ (which we write as $f_{n,P}(Q)$ in the sequel). This is done using Miller's algorithm described in the next subsection.

2.2 Computation of Pairings and Miller's Algorithm

In order to emphasize that our improvement can be applied in a very general context, we explain briefly in this subsection how pairings are computed. In what follows, $\mathbb{F}$ denotes a field (not necessarily finite), E an elliptic curve over $\mathbb{F}$ and r an integer not divisible by the characteristic of $\mathbb{F}$. We suppose that the group $E(\mathbb{F})$ of $\mathbb{F}$-rational points of E contains a point P of order r. Since r is prime to the characteristic of $\mathbb{F}$, the group $E[r]$ of points of order r of E is isomorphic to a direct sum of two cyclic groups of order r. In general, a point $Q \in E[r]$ that is not a multiple of P will be defined over some extension $\mathbb{F}'$ of $\mathbb{F}$ of finite degree.

If P, P' are two points in $E(\mathbb{F})$, we denote by $\ell_{P,P'}$ a function with divisor $[P] + [P'] + [-(P + P')] - 3[O_E]$ and by v_P a function with divisor $[P]+[-P]-2[O_E]$. Clearly these functions are only defined up to a multiplicative constant; we recall at the end of this section how to normalize functions so that they are uniquely determined by their divisor. Note that v_P is just the same as $\ell_{P,-P}$.

If s and t are two integers, we denote by $f_{s,P}$ (or simply f_s if there is no possibility of confusion) a function whose divisor is $s[P]-[sP]-(s-1)[O_E]$. We abbreviate $\ell_{sP,tP}$ to $\ell_{s,t}$ and v_{sP} to v_s. As understood in this paper, the purpose of Miller's algorithm is to calculate $f_{s,P}(Q)$ when $Q \in E[r]$ is not a multiple of P. All pairings can be expressed in terms of these functions for appropriate values of s.

Miller's algorithm is based on the following Lemma describing the so-called *Miller's formula*, which is proved by considering divisors.

Lemma 1. *For s and t two integers, up to a multiplicative constant, we have $f_{s+t} = f_s f_t \frac{\ell_{s,t}}{v_{s+t}}$.*

The usual Miller algorithm makes use of Lemma 1 with $t = s$ in a doubling step and $t = 1$ in an addition step. It is described by the pseudocode in Figure 1, which presents the algorithm updating numerators and denominators separately, so that just one inversion is needed at the end. We write the functions ℓ and v as quotients $(N\ell)/(D\ell)$ and $(Nv)/(Dv)$, where each of the terms $(N\ell)$, $(D\ell)$, (Nv), (Dv) is computed using only additions and multiplications, and no inversions. Here the precise definitions of $(N\ell)$, $(D\ell)$, (Nv), (Dv) will depend on the representations that are used; in Section 3.2 we indicate one such choice when short Weierstrass coordinates and the associated Jacobian coordinates are used. In the algorithm, T is always a multiple of P, so that the hypothesis that Q is not a multiple of P implies that at the functions $\ell_{T,T}$, $\ell_{T,P}$, v_{2T} and v_{T+P}

cannot vanish at Q. It follows that f and g never vanish at Q so that the final quotient f/g is well-defined and non-zero.

Algorithm 1: Miller(P, Q, s) usual

Data: $s = \sum_{i=0}^{l-1} s_i 2^i$ (radix 2), $s_i \in \{0, 1\}$, $Q \in E(\mathbb{F}')$ not a multiple of P.
Result: $f_{s,P}(Q)$.
$T \leftarrow P$, $f \leftarrow 1$, $g \leftarrow 1$,
for $i = l - 2$ **to** 0 **do**
  $f \leftarrow f^2 (N\ell)_{T,T}(Dv)_{2T}$,
  $g \leftarrow g^2 (D\ell)_{T,T}(Nv)_{2T}$,
  $T \leftarrow 2T$
  if $s_i = 1$ **then**
    $f \leftarrow f(N\ell)_{T,P}(Dv)_{T+P}$,
    $g \leftarrow g(D\ell)_{T,P}(Nv)_{T+P}$,
    $T \leftarrow T + P$
  end
end
return f/g

Fig. 1. The usual Miller algorithm

Obviously in any implementation it is essential for the functions appearing in the programs to be uniquely determined, and so we end this section by recalling briefly how this can be done in our context. If w is a uniformizer at O_E, we say that the non-zero rational function f on E is normalized (or monic) with respect to w if the Laurent expansion of f at O_E is of the form

$$f = w^n + c_{n+1} w^{n+1} + c_{n+2} w^{n+2} + \cdots, \quad c_i \in \mathbb{F},$$

(i. e. if the first non-zero coefficient is 1). Any non-zero rational function on E can be written in a unique way as a product of a constant and a normalized function, and the normalized rational functions form a group under multiplication that is isomorphic to the group of principal divisors.

As a typical example, when E is in short Weierstrass form

$$y^2 = x^3 + ax + b, \ a, b \in \mathbb{F} \tag{2}$$

one can take $w = \frac{x}{y}$. Any rational function on E can be written as a quotient of two functions whose polar divisors are supported at the origin O_E of E. If f is a function whose only pole is at O_E, then there exist two polynomials $U(x)$ and $V(x)$ such that $f = U(x) + V(x)y$: here U and V are uniquely determined by f. Furthermore, if the order of the pole of f is n, then $n \geq 2$ and $U(x)$ is of degree $\frac{n}{2}$ and $V(x)$ of degree at most $\frac{n-4}{2}$ if n is even and $V(x)$ is of degree $\frac{n-3}{2}$ and $U(x)$ of degree at most $\frac{n-1}{2}$ when n is odd. Then f is normalized if and only if,

when f is written in the form $U(x)+V(x)y$, $U(x)$ is monic when n is even and $V(x)$ is monic when n is odd. In general, a rational function on E is normalized if and only if it is a quotient of two normalized functions whose polar divisors are supported by the origin.

For example, when E is given by the equation (2), the functions

$$\ell_{T,P}(x,y) = \begin{cases} y - y_P - \frac{y_T - y_P}{x_T - x_P}(x - x_P), & T \neq O_E,\ \pm P, \\ y - y_P - \frac{3x_P^2 + a}{2y_P}(x - x_P), & T = P,\ 2P \neq O_E. \end{cases}$$

and

$$v_P(x,y) = x - x_P$$

are normalized, so that if they are used in implementations of the algorithms given in Figures 1 and 2, then these algorithms output $f_{s,P}(Q)$ with $f_{s,P}$ the normalized function with divisor $s[P] - [sP] - (s-1)[O_E]$.

3 Our Variant of Miller's Algorithm

In this section, we describe our variant of Miller's formula and algorithm and analyze the cost of the latter in terms of basic operations.

3.1 The Algorithm

The main improvement comes from the following Lemma.

Lemma 2. *For s and t two integers, up to a multiplicative constant, we have*

$$f_{s+t} = \frac{1}{f_{-s} f_{-t} \ell_{-s,-t}}.$$

Proof. This lemma is again proved by considering divisors. Indeed,

$$\begin{aligned} \mathrm{div}(f_{-s} f_{-t} \ell_{-s,-t}) &= (-s)[P] - [(-s)P] + (s+1)[O_E] \\ &\quad + (-t)[P] - [(-t)P] + (t+1)[O_E] \\ &\quad + [-sP] + [-tP] + [(s+t)P] - 3[O_E] \\ &= -(s+t)[P] + [(s+t)P] + (s+t-1)[O_E] \\ &= -\mathrm{div}(f_{s+t}), \end{aligned}$$

which concludes the proof. □

We shall seek to exploit the fact that here the right hand member has only three terms whereas that of Lemma 1 has four.

Our variant of Miller's algorithm is described by the pseudocode in Figure 2. It was inspired by the idea of applying Lemma 2 with $t = s$ or $t \in \{\pm 1\}$. However, the scalar input is given in binary representation. It updates numerators and denominators separately, so that only one final inversion appears at the end. As

in Figure 1, the hypothesis that Q is not a multiple of P implies that at no stage do f and g vanish, so that the final quotient f/g makes sense. Note that T is always a *positive* multiple of P. We use the notation $\ell'_{-T,-P}$ for the function $f_{-1}\ell_{-T,-P}$, since in many situations it can be computed faster than simply by computing f_{-1} and $\ell_{-T,-P}$ and taking the product. For example, when E is given in short Weierstrass coordinates by the equation $y^2 = x^3 + ax + b$, a, $b \in \mathbb{F}$, we have

$$\ell'_{-T,-P} = f_{-1}\ell_{-T,-P} = \frac{1}{x_Q - x_P}\big(y_Q + y_P + \lambda(x_Q - x_P)\big) = \frac{y_Q + y_P}{x_Q - x_P} + \lambda, \quad (3)$$

where $\frac{y_Q+y_P}{x_Q-x_P}$ can be precomputed (at the cost of one inversion and one multiplication in the big field) and λ denotes the slope of the line joining T to P.

The tables in Figures 3 and 4 show that our variant is more efficient than the classical Miller's algorithm as we save a product in the big field at each doubling and each addition step. We also save some multiplications and squarings in $\mathbb{F}$. The following subsection discusses all this in more detail. In Section 4 we describe a version without denominators that works for elliptic curves with even embedding degree.

3.2 Generic Analysis

In this subsection, we compare the number of operations needed to compute $f_{s,P}(Q)$ using the algorithms in Figures 1 and 2. In order to fix ideas, we make our counts using Jacobian coordinates (X, Y, Z) associated to a short Weierstrass model $y^2 = x^3 + ax + b$, a, $b \in \mathbb{F}$, so that $x = X/Z^2$ and $y = Y/Z^3$. We suppose that the Jacobian coordinates of P lie in $\mathbb{F}$ and that those of Q lie in some extension $\mathbb{F}'$ of $\mathbb{F}$ of whose degree is denoted by k. We denote by $\mathbf{m}_a$ the multiplication by the curve coefficient a and we denote respectively by $\mathbf{m}$ and $\mathbf{s}$ multiplications and squares in $\mathbb{F}$, while the same operations in $\mathbb{F}'$ are denoted respectively by $\mathbf{M}_k$ and $\mathbf{S}_k$ if k is the degree of the extension $\mathbb{F}'$. We assume that $\mathbb{F}'$ is given by a basis as a $\mathbb{F}$-vector space one of whose elements is 1, so that multiplication of an element of $\mathbb{F}'$ by an element of $\mathbb{F}$ counts as k multiplications in $\mathbb{F}$. We ignore additions and multiplications by small integers.

If S is any point of E, then X_S, Y_S and Z_S denote the Jacobian coordinates of S, so that when $S \neq O_E$, the Weierstrass coordinates of S are $x_S = X_S/Z_S^2$ and $y_S = Y_S/Z_S^3$. As before, T is a multiple of P, so that X_T, Y_T and Z_T all lie in $\mathbb{F}$. Since P and Q are part of the input, we assume they are given in Weierstrass coordinates and that $Z_P = Z_Q = 1$.

We need to define the numerators and the denominators of the quantities appearing in the algorithms of Figures 1 and 2. The cost of computing these quantities and the total cost of these algorithms are analyzed in Figures 3 and 4.

Algorithm 2: Miller(P, Q, r) modified

Data: $s = \sum_{i=0}^{l-1} s_i 2^i$, $s_i \in \{0, 1\}$, $s_{l-1} = 1$, h Hamming weight of s, $Q \in E(\mathbb{F}')$ not a multiple of P
Result: $f_{s,P}(Q)$;
$f \leftarrow 1$, $T \leftarrow P$,
if $l + h$ *is odd* **then**
  $\delta \leftarrow 1$, $g \leftarrow f_{-1}$
end
else
  $\delta \leftarrow 0$, $g \leftarrow 1$
end
for $i = l - 2$ **to** 0 **do**
1 **if** $\delta = 0$ **then**
    $f \leftarrow f^2 (N\ell)_{T,T}$,
    $g \leftarrow g^2 (D\ell)_{T,T}$,
    $T \leftarrow 2T$, $\delta \leftarrow 1$
2   **if** $s_i = 1$ **then**
      $g \leftarrow g(N\ell')_{-T,-P}$,
      $f \leftarrow f(D\ell')_{-T,-P}$,
      $T \leftarrow T + P$, $\delta \leftarrow 0$
    end
  end
3 **else**
    $g \leftarrow g^2 (N\ell)_{-T,-T}$,
    $f \leftarrow f^2 (D\ell)_{-T,-T}$,
    $T \leftarrow 2T$, $\delta \leftarrow 0$
4   **if** $s_i = 1$ **then**
      $f \leftarrow f(N\ell)_{T,P}$,
      $g \leftarrow g(D\ell)_{T,P}$,
      $T \leftarrow T + P$, $\delta \leftarrow 1$
    end
  end
end
return f/g

Fig. 2. Our modified Miller algorithm

The doubling step. We first deal with doubling. Suppose that $2T \neq O_E$, which will always be the case if r is odd. Then $y_T \neq 0$ and the slope of the tangent to E at T is

$$\mu_T = \frac{3x_T^2 + a}{2y_T} = \frac{(N\mu)_T}{(D\mu)_T},$$

where $(N\mu)_T = 3X_T^2 + aZ_T^4$ and $(D\mu)_T = 2Y_TZ_T = (Y_T + Z_T)^2 - Y_T^2 - Z_T^2$. Hence the value of $\ell_{T,T}$ at Q can be written

$$\ell_{T,T}(Q) = y_Q - y_T - \mu_T(x_Q - x_T) = \frac{(N\ell)_{T,T}}{(D\ell)_{T,T}},$$

where now

$$(N\ell)_{T,T} = (D\mu)_T Z_T^2 y_Q - (N\mu)_T Z_T^2 x_Q - 2Y_T^2 - (N\mu)_T X_T$$

and

$$(D\ell)_{T,T} = (D\mu)_T Z_T^2.$$

The coordinates X_{2T}, Y_{2T} and Z_{2T} of $2T$ are given by

$$\begin{cases} X_{2T} = (N\mu)_T^2 - 8X_TY_T^2, \\ Y_{2T} = (N\mu)_T(4X_TY_T^2 - X_{2T}) - 8Y_T^4, \\ Z_{2T} = (D\mu)_T. \end{cases}$$

Hence the value of v_{2T} at Q can be calculated as

$$v_{2T}(Q) = x_Q - x_{2T} = \frac{(Nv)_{2T}}{(Dv)_{2T}},$$

where $(Dv)_{2T} = Z_{2T}^2$ and $(Nv)_{2T} = Z_{2T}^2 x_Q - X_{2T}$.

In the modified algorithm, we also need to compute $\ell_{-T,-T}$ at Q. But the coordinates of $-T$ are $(x_T, -y_T)$, so that we can write

$$\ell_{-T,-T}(Q) = y_Q + y_T + \mu_T(x_Q - x_T) = \frac{(N\ell)_{-T,-T}}{(D\ell)_{-T,-T}},$$

with

$$(N\ell)_{-T,-T} = (D\mu)_T Z_T^2 y_Q + (N\mu)_T Z_T^2 x_Q + 2Y_T^2 - (N\mu)_T X_T$$

and

$$(D\ell)_{-T,-T} = (D\mu)_T Z_T^2.$$

All the operations needed in the doubling steps of the algorithms in Figures 1 and 2 are shown in detail in Figure 3. The quantities are to be computed in the order shown in the table, a blank entry indicating that the corresponding quantity need not be computed in the case indicated at the top of the corresponding column. The costs of the computations are calculated assuming that the results of intermediate steps are kept in memory. An entry 0 indicated that the quantity has already been calculated at a previous stage.

Quantity	Formula	Classical Miller (Fig. 1)	Modified Miller (Fig. 2, loop **1**)	Modified Miller (Fig. 2, loop **3**)
$(N\mu)_T$	$3X_T^2 + aZ_T^4$	$\mathbf{m}_a + 3\mathbf{s}$	$\mathbf{m}_a + 3\mathbf{s}$	$\mathbf{m}_a + 3\mathbf{s}$
$(D\mu)_T$	$2Y_T Z_T = (Y_T + Z_T)^2 - Y_T^2 - Z_T^2$	$2\mathbf{s}$	$2\mathbf{s}$	$2\mathbf{s}$
$(N\ell)_{T,T}$	$(D\mu)_T Z_T^2 y_Q - (N\mu)_T Z_T^2 x_Q$ $-2Y_T^2 - (N\mu)_T X_T$	$(3+2k)\mathbf{m}$	$(3+2k)\mathbf{m}$	
$(D\ell)_{T,T}$	$(D\mu)_T Z_T^2$	0	0	
X_{2T}	$(N\mu)_T^2 - 8X_T Y_T^2$	$\mathbf{s} + \mathbf{m}$	$\mathbf{s} + \mathbf{m}$	$\mathbf{s} + \mathbf{m}$
Y_{2T}	$(N\mu)_T(4X_T Y_T^2 - X_{2T}) - 8Y_T^4$	$\mathbf{s} + \mathbf{m}$	$\mathbf{s} + \mathbf{m}$	$\mathbf{s} + \mathbf{m}$
Z_{2T}	$(D\mu)_T$	0	0	0
$(Dv)_{2T}$	Z_{2T}^2	$\mathbf{s}$		
$(Nv)_{2T}$	$Z_{2T}^2 x_Q - X_{2T}$	$k\mathbf{m}$		
$(N\ell)_{-T,-T}$	$(D\mu)_T Z_T^2 y_Q + (N\mu)_T Z_T^2 x_Q$ $+2Y_T^2 - (N\mu)_T X_T$			$(3+2k)\mathbf{m}$
$(D\ell)_{-T,-T}$	$(D\mu)_T Z_T^2$			0
f	$\leftarrow\ f^2 (N\ell)_{T,T}(Dv)_{T,T}$	$k\mathbf{m} + \mathbf{S}_k + \mathbf{M}_k$		
g	$\leftarrow\ g^2 (D\ell)_{T,T}(Nv)_{T,T}$	$k\mathbf{m} + \mathbf{S}_k + \mathbf{M}_k$		
f	$\leftarrow\ f^2 (N\ell)_{T,T}$		$\mathbf{S}_k + \mathbf{M}_k$	
g	$\leftarrow\ g^2 (D\ell)_{T,T}$		$k\mathbf{m} + \mathbf{S}_k$	
f	$\leftarrow\ f^2 (D\ell)_{-T,-T}$			$k\mathbf{m} + \mathbf{S}_k$
g	$\leftarrow\ g^2 (N\ell)_{-T,-T}$			$\mathbf{S}_k + \mathbf{M}_k$
TOTAL		$\mathbf{m}_a + 8\mathbf{s}$ $+(5+5k)\mathbf{m}$ $+2\mathbf{S}_k + 2\mathbf{M}_k$	$\mathbf{m}_a + 7\mathbf{s}$ $+(5+3k)\mathbf{m}$ $+2\mathbf{S}_k + \mathbf{M}_k$	$\mathbf{m}_a + 7\mathbf{s}$ $+(5+3k)\mathbf{m}$ $+2\mathbf{S}_k + \mathbf{M}_k$

Fig. 3. Analysis of the cost of generic doubling

The addition step. Next we deal with addition. When $T \neq \pm P$, the slope of the line joining T and P is

$$\lambda_{T,P} = \frac{y_T - y_P}{x_T - x_P} = \frac{(N\lambda)_{T,P}}{(D\lambda)_{T,P}},$$

where $(N\lambda)_{T,P} = Y_T - y_P Z_T^3$ and $(D\lambda)_{T,P} = X_T Z_T - x_P Z_T^3$.

It follows that the value of $\ell_{T,P}$ at the point Q is given by

$$y_Q - y_P - \lambda(x_Q - x_P) = \frac{(N\ell)_{T,P}}{(D\ell)_{T,P}},$$

where we precompute $y_Q - y_P$ and $x_Q - x_P$, and the numerator and denominator are given by

$$(N\ell)_{T,P} = (D\lambda)_{T,P}(y_Q - y_P) - (N\lambda)_{T,P}(x_Q - x_P)$$

and

$$(D\ell)_{T,P} = (D\lambda)_{T,P}.$$

The coordinates X_{T+P}, Y_{T+P} and Z_{T+P} of $T + P$ are then given by

$$\begin{cases} X_{T+P} = (N\lambda)_{T,P}^2 - (X_T + x_P Z_T^2)(X_T - x_P Z_T^2)^2, \\ Y_{T+P} = -(D\lambda)_{T,P}^3 y_P + (N\lambda)_{T,P}(x_P (D\lambda)_{T,P}^2 - X_{T+P}), \\ Z_{T+P} = (D\lambda)_{T,P}. \end{cases}$$

It follows that we can write the value of v_{T+P} at Q as

$$v_{T+P}(Q) = x_Q - x_{T+P} = \frac{(Nv)_{T+P}}{(Dv)_{T+P}},$$

with $(Nv)_{T+P} = x_Q Z_{T+P}^2 - X_{T+P}$ and $(Dv)_{T+P} = Z_{T+P}^2$.

When the loop beginning with line **2** of Figure 2 is used, we need to calculate $\ell'_{-T,-P}$ at Q. In fact, using equation (3), we can write

$$\ell'_{-T,-P}(Q) = \frac{y_Q + y_P}{x_Q - x_P} + \frac{(N\lambda)_{T,P}}{(D\lambda)_{T,P}}.$$

If $\frac{y_Q+y_P}{x_Q-x_P}$ has been precomputed and has value $\alpha_{Q,P}$, we can write

$$\ell'_{-T,-P}(Q) = \frac{(N\ell')_{-T,-P}}{(D\ell')_{-T,-P}},$$

with $(N\ell')_{-T,-P} = (D\lambda)_{T,P}\alpha_{Q,P} + (N\lambda)_{T,P}$ and $(D\ell')_{-T,-P} = (D\lambda)_{T,P}$.

All the operations needed in the addition steps of the algorithms in Figures 1 and 2 are shown in detail in Figure 4. As with the doubling step, the quantities are to be computed in the order shown in the table, a blank entry indicating that the corresponding quantity need not be computed in the case indicated at the top of the corresponding column. The costs of the computations are calculated assuming that the results of intermediate steps are kept in memory. An entry 0 indicated that the quantity has already been calculated at a previous stage.

3.3 The main result

The following theorem recapitulates the number of operations in our variant of Miller's algorithm.

Theorem 2. *Suppose E is given in short Weierstrass form $y^2 = x^3 + ax + b$ with coefficients a, $b \in \mathbb{F}$. Let $P \in E(\mathbb{F})$ be a point of odd order $r \geq 2$ and let Q be a point of E of order r with coordinates in an extension field $\mathbb{F}'$ of $\mathbb{F}$ of degree k. We assume P and Q given in Weierstrass coordinates (x_P, y_P) and (x_Q, y_Q).*

1. *Using the associated Jacobian coordinates, the algorithms of Figures 1 and 2 can be implemented in such a way that all the denominators $(D\ell)_{T,T}$, $(D\ell)_{T,P}$, $(Dv)_{2T}$, $(Dv)_{T+P}$ and $(D\ell')_{-T,-P}$ belong to $\mathbb{F}$.*
2. *When this is the case:*
   (a) *Each doubling step of the generic usual Miller algorithm takes $\mathbf{m}_a + 8\mathbf{s} + (5+5k)\mathbf{m} + 2\mathbf{S}_k + 2\mathbf{M}_k$ operations while in the generic modified Miller algorithm it requires only $\mathbf{m}_a + 7\mathbf{s} + (5+3k)\mathbf{m} + 2\mathbf{S}_k + \mathbf{M}_k$ operations.*
   (b) *Each addition step of the generic usual Miller algorithm takes $4\mathbf{s} + (8+5k)\mathbf{m} + 2\mathbf{M}_k$ operations. On the other hand, the generic modified Miller algorithm requires only $3\mathbf{s} + (8+2k)\mathbf{m} + \mathbf{M}_k$ operations when line **2** is needed and $3\mathbf{s} + (8+3k)\mathbf{m} + \mathbf{M}_k$ operations when line **4** is needed.*

Quantity	Formula	Classical Miller (Fig. 1)	Modified Miller (Fig. 2, loop **2**)	Modified Miller (Fig. 2, loop **4**)
$(D\lambda)_{T,P}$	$(X_T - x_P Z_T^2)Z_T$	$\mathbf{s} + 2\mathbf{m}$	$\mathbf{s} + 2\mathbf{m}$	$\mathbf{s} + 2\mathbf{m}$
$(N\lambda)_{T,P}$	$Y_T - y_P Z_T^3$	$2\mathbf{m}$	$2\mathbf{m}$	$2\mathbf{m}$
$(N\ell)_{T,P}$	$(D\lambda)_{T,P}(y_Q - y_P)$ $-(N\lambda)_{T,P}(x_Q - x_P)$	$2k\mathbf{m}$		$2k\mathbf{m}$
$(D\ell)_{T,P}$	$(D\lambda)_{T,P}$	0		0
X_{T+P}	$(N\lambda)_{T,P}^2 - X_T(X_T - x_P Z_T^2)^2$ $-x_P Z_T^2(X_T - x_P Z_T^2)^2$	$2\mathbf{s} + 2\mathbf{m}$	$2\mathbf{s} + 2\mathbf{m}$	$2\mathbf{s} + 2\mathbf{m}$
Y_{T+P}	$-y_P Z_T^3(X_T(X_T - x_P Z_T^2)^2$ $-x_P Z_T^2(X_T - x_P Z_T^2)^2)$ $+(N\lambda)_{T,P}(x_P Z_T^2(X_T - x_P Z_T^2)^2$ $-X_{T+P})$	$2\mathbf{m}$	$2\mathbf{m}$	$2\mathbf{m}$
Z_{T+P}	$(D\lambda)_{T,P}$	0	0	0
$(Nv)_{T+P}$	$x_Q Z_{T+P}^2 - X_{T+P}$	$\mathbf{s} + k\mathbf{m}$		
$(Dv)_{T+P}$	Z_{T+P}^2	0		
$(N\ell')_{-T,-P}$	$\alpha_{Q,P}(D\lambda)_{T,P} + (N\lambda)_{T,P}$		$k\mathbf{m}$	
$(D\ell')_{-T,-P}$	$(D\lambda)_{T,P}$		0	
f	$\leftarrow\ f(N\ell)_{T,P}(Dv)_{T+P}$	$k\mathbf{m} + \mathbf{M}_k$		
g	$\leftarrow\ g(D\ell)_{T,P}(Nv)_{T+P}$	$k\mathbf{m} + \mathbf{M}_k$		
f	$\leftarrow\ f(D\ell')_{-T,-P}$		$k\mathbf{m}$	
g	$\leftarrow\ g(N\ell')_{-T,-P}$		$\mathbf{M}_k$	
f	$\leftarrow\ f(N\ell)_{T,P}$			$\mathbf{M}_k$
g	$\leftarrow\ g(D\ell)_{T,P}$			$k\mathbf{m}$
TOTAL		$4\mathbf{s} + (8 + 5k)\mathbf{m}$ $+2\mathbf{M}_k$	$3\mathbf{s} + (8 + 2k)\mathbf{m}$ $+\mathbf{M}_k$	$3\mathbf{s} + (8 + 3k)\mathbf{m}$ $+\mathbf{M}_k$

Fig. 4. Analysis of the cost of generic addition

We have made no serious attempt to minimize the number of operations, for example by using formulas similar to those in [1].

Since the first part of Theorem 2 implies that, when $\mathbb{F}$ is a finite field with q elements, we have

$$(D\ell)_{T,T}^{q-1} = (D\ell)_{T,P}^{q-1} = (Dv)_{2T}^{q-1} = (Dv)_{T+P}^{q-1} = (D\ell')_{-T,-P}^{q-1} = 1,$$

denominator elimination is possible when we only need to calculate $f_{s,P}(Q)$ to some power divisible by $q-1$. Such an algorithm saves at least $k\mathbf{m}$ operations both in the classical version and our variant.

Our new version of Miller's algorithm works perfectly well for arbitrary embedding degree. For example, using Theorem 2 of [16], it should be possible to find an elliptic curve with a prime embedding degree minimizing the number of iterations. *Optimal pairings* [26] involve in their computation a product $\prod_{i=0}^{\ell} f_{c_i,Q}^{q^i}(P)$ whose terms can be computed with our algorithm. Note that switching P and Q will lead to more computations in the extension field, but it is shown in [23] that optimized versions of the Ate and the twisted Ate pairing can be computed at least as fast as the Tate pairing. Note that Heß [16] §5, also mentions pairings of potential interest when k is odd and the elliptic curve has discriminant -4 and when k is not divisible by 3 and the elliptic curve has discriminant -3.

4 Curves with even embedding degree

Currently, most implementations (where $\mathbb{F}$ is a finite field) are adapted to curves with embedding degree $2^i 3^j$, since the usual version of Miller's algorithm can be implemented more efficiently. Indeed, such curves admit an even twist which allows denominator elimination [4, 5]. In the case of a cubic twist, denominator elimination is also possible [22]. Another advantage of embedding degrees of the form $2^i 3^j$ is that the corresponding extensions of $\mathbb{F}$ can be written as composite extensions of degree 2 or 3, which allows faster basic arithmetic operations [21].

In what follows, we discuss a version with denominator elimination of our variant adapted to even embedding degrees. Similar ideas have been used before, for example in [22] or [13]. We suppose that $\mathbb{F}$ is a finite field of odd cardinality q which we denote by $\mathbb{F}_q$. We consider an elliptic curve E with short Weierstrass model $y^2 = x^3 + ax + b$ with $a, b \in \mathbb{F}_q$. We suppose that $E(\mathbb{F}_q)$ contains a point P with affine coordinates (x_P, y_P) of order an odd prime r and embedding degree k. If $n \geq 1$ is an integer, we denote by $\mathbb{F}_{q^n}$ the extension of degree n of $\mathbb{F}_q$ in a fixed algebraic closure of $\mathbb{F}_q$.

We suppose for the rest of this section that k is even. Let γ be a non-square element of $\mathbb{F}_{q^{k/2}}$ and fix a square root β in $\mathbb{F}_{q^k}$ of γ, so that every element of $\mathbb{F}_{q^k}$ can be written in a unique way in the form $x + y\beta$ with $x, y \in \mathbb{F}_{q^{k/2}}$. Then one knows that $E[r]$ contains non zero points $Q = (x_Q, y_Q)$ that satisfy x_Q, $y_Q/\beta \in \mathbb{F}_{q^{k/2}}$. In fact, if π denotes the Frobenius endomorphism of E over $\mathbb{F}_q$ given by $\pi(x, y) = (x^q, y^q)$, then π restricts to an endomorphism of $E[r]$ viewed as a vector space over the field with r elements, and a point $Q \in E[r] \setminus \{O_E\}$ has the desired property if and only if Q belongs to the eigenspace V of π with eigenvector q. We can easily construct such points Q as follows: if $R \in E[r]$, then the point

$$Q := [k](R) - \sum_{i=0}^{k-1} \pi^i(R)$$

lies in V and, when r is large, the probability that it is non zero when R is selected at random is overwhelming.

As is well-known, this leads to speed-ups in the calculation of $f_{s,P}(Q)$ whenever $Q \in V$. For example, the calculation of $(N\ell)_{T,P}$ (see Figure 4) takes only $(2\frac{k}{2})\mathbf{m} = k\mathbf{m}$ operations. On the other hand, in general neither $\alpha_{Q,P}$ nor $\alpha_{Q,P}/\beta$ belongs to $\mathbb{F}_{q^{k/2}}$, so that the calculation of $(N\ell')_{-T,-P}$ also requires $k\mathbf{m}$ operations. So, the improvement obtained in the generic case is lost.

When $P \in E[r](\mathbb{F}_q)$ and $Q \in V$, it is well-known that the Tate pairing $e_r^T(P, Q)$ is given by $e_r^T(P, Q) = f_{r,P}(Q)^{\frac{q^k-1}{r}}$. Denominator elimination is possible since $q^{k/2} - 1$ divides $\frac{q^k-1}{r}$, as follows from the fact that r is prime and the definition of the embedding degree k.

Let $v = x + y\beta$ with $x, y \in \mathbb{F}_{q^{k/2}}$ be an element of $\mathbb{F}_{q^k}$. The conjugate of v over $\mathbb{F}_{q^{k/2}}$ is then $\bar{v} = x - y\beta$. It follows that, if $v \neq 0$, then

$$\frac{1}{v} = \frac{\overline{v}}{x^2 - \gamma y^2} \tag{4}$$

where $x^2 - \gamma y^2 \in \mathbb{F}_{q^{k/2}}$. Thus, in a situation where elements of $\mathbb{F}_{q^{k/2}}$ can be ignored, $\frac{1}{v}$ can be replaced by $\overline{v}$, thereby saving an inversion in $\mathbb{F}_{q^k}$.

We exploit all this in the algorithm in Figure 5, where we use the same notation as in Figure 2. It outputs an element f of $\mathbb{F}_{q^k}$ such that $f^{q^{\frac{k}{2}}-1} = f_{s,P}(Q)^{q^{\frac{k}{2}}-1}$. Thus, when $s = r$, we find, since $q^{\frac{k}{2}} - 1$ divides $\frac{q^k-1}{r}$, that $e_r^T(P,Q) = f^{\frac{q^k-1}{r}}$.

We replace the denominators $\ell_{-T,-T}$ and $\ell_{-T,-P}$ (updated in the function g) by their conjugates $\overline{\ell_{-T,-T}}$ and $\overline{\ell_{-T,-P}}$. In Jacobian coordinates, one has

$$\overline{(N\ell')_{-T,-P}} = \overline{\alpha_{Q,P}}(D\lambda)_{T,P} + (N\lambda)_{T,P}, \qquad \text{and} \tag{5}$$

$$\overline{(N\ell)_{-T,-T}} = 2Y_T(-y_Q Z_T^3 + Y_T) + (N\mu)_T(x_Q Z_T^2 - X_T). \tag{6}$$

Cost analysis of the doubling and addition steps in this algorithm can be found in Figures 6 and 7. We summarize our conclusions in the following result.

Theorem 3. *Let β be a non-square element of $\mathbb{F}_{q^k}$ whose square lies in $\mathbb{F}_{q^{k/2}}$. Suppose E is given in short Weierstrass form $y^2 = x^3 + ax + b$ with coefficients a, $b \in \mathbb{F}_q$. Let $P \in E(\mathbb{F}_q)$ be a point of odd prime order r with even embedding degree k. Let Q be a point of E of order r with coordinates in the extension field $\mathbb{F}_{q^k}$ of $\mathbb{F}_q$ of degree k. We assume P and Q given in Weierstrass coordinates (x_P, y_P) and (x_Q, y_Q) with x_Q, $y_Q/\beta \in \mathbb{F}_{q^{k/2}}$. Using the associated Jacobian coordinates, every doubling step of the algorithm of Figure 5 requires $\mathbf{m}_a + 7\mathbf{s} + (5+k)\mathbf{m} + \mathbf{S}_k + \mathbf{M}_k$ operations and every addition step requires $3\mathbf{s} + (8+k)\mathbf{m} + \mathbf{M}_k$.*

As before, we have made no serious attempt to minimize the number of operations, for example by using formulas similar to those in [1]. Moreover, we believe that there is scope for further improvement in the case of curves in special families or curves with efficient arithmetic (see for example [14]).

5 Experiments

We ran some experiments comparing usual Miller (Figure 1) with the variant of Figure 2 when $k = 17$ and $k = 19$. When $k = 18$, we compared the performance of the algorithms of Figures 1 and 5 and also the algorithm proposed in 2003 by [4]. In each case, the group order r has 192 bits and the rho-value $\rho = \frac{\log q}{\log r}$ is a little under 1.95, q being the cardinality of the base field. In the example with $k = 17$, the big field $\mathbb{F}_{q^k}$ was generated as $\mathbb{F}_q[x]/(x^{17}+x+12)$ while when $k = 18$ and $k = 19$, $\mathbb{F}_{q^k}$ was generated as $\mathbb{F}_q[x]/(x^k + 2)$. Our curves were constructed using the Cocks-Pinch method (see [15]).

– For $k = 17$, the curve is $E : y^2 = x^3 + 6$ and

$r = 627710173538668076383578942320766641610235544446403993985 7$

$q = 220522060437535156222191257592633526736200793713321924733$
$078476278317592333015347957276809006053649122618843827 71$

Algorithm 3: Miller(P, Q, s) modified with even embedding degree

Data: $s = \sum_{i=0}^{l-1} s_i 2^i$, $s_i \in \{0,1\}$, $s_{l-1} = 1$, h Hamming weight of s, $Q \in E[r]$ a non-zero element with x_Q, $y_Q/\beta \in \mathbb{F}_{q^{k/2}}$.

Result: An element f of $\mathbb{F}_{q^k}$ satisfying $f^{q^{k/2}-1} = f_{s,P}(Q)^{q^{k/2}-1}$

$f \leftarrow 1$, $T \leftarrow P$,
if $l + h$ *is odd* **then**
  $\delta \leftarrow 1$
end
else
  $\delta \leftarrow 0$
end
for $i = l - 2$ **to** 0 **do**
1 **if** $\delta = 0$ **then**
    $f \leftarrow f^2 (N\ell)_{T,T}$, $T \leftarrow 2T$, $\delta \leftarrow 1$
2   **if** $s_i = 1$ **then**
      $f \leftarrow f\overline{(N\ell')_{-T,-P}}$, $T \leftarrow T + P$, $\delta \leftarrow 0$
    end
  end
3 **else**
    $f \leftarrow f^2\overline{(N\ell)_{-T,-T}}$, $T \leftarrow 2T$, $\delta \leftarrow 0$
4   **if** $s_i = 1$ **then**
      $f \leftarrow f(N\ell)_{T,P}$, $T \leftarrow T + P$, $\delta \leftarrow 1$
    end
  end
end
return f

Fig. 5. The modified Miller algorithm for even embedding degree

Quantity	Formula	Modified Miller (Fig. 5, loop **1**)	Modified Miller (Fig. 5, loop **3**)
$(N\mu)_T$	$3X_T^2 + aZ_T^4$	$\mathbf{m}_a + 3\mathbf{s}$	$\mathbf{m}_a + 3\mathbf{s}$
$(D\mu)_T$	$2Y_T Z_T = (Y_T + Z_T)^2 - Y_T^2 - Z_T^2$	$2\mathbf{s}$	$2\mathbf{s}$
$(N\ell)_{T,T}$	$(D\mu)_T Z_T^2 y_Q - (N\mu)_T Z_T^2 x_Q - 2Y_T^2 - (N\mu)_T X_T$	$(3 + 2(k/2))\mathbf{m}$	
X_{2T}	$(N\mu)_T^2 - 8X_T Y_T^2$	$\mathbf{s} + \mathbf{m}$	$\mathbf{s} + \mathbf{m}$
Y_{2T}	$(N\mu)_T(4X_T Y_T^2 - X_{2T}) - 8Y_T^4$	$\mathbf{s} + \mathbf{m}$	$\mathbf{s} + \mathbf{m}$
Z_{2T}	$(D\mu)_T$	0	0
$(N\ell)_{-T,-T}$	$(D\mu)_T Z_T^2 y_Q + (N\mu)_T Z_T^2 x_Q + 2Y_T^2 - (N\mu)_T X_T$		$(3 + 2(k/2))\mathbf{m}$
f	$\leftarrow f^2 (N\ell)_{T,T}$	$\mathbf{S}_k + \mathbf{M}_k$	
f	$\leftarrow f^2\overline{(N\ell)_{-T,-T}}$		$\mathbf{S}_k + \mathbf{M}_k$
TOTAL		$\mathbf{m}_a + 7\mathbf{s}$ $+(5+k)\mathbf{m}$ $+\mathbf{S}_k + \mathbf{M}_k$	$\mathbf{m}_a + 7\mathbf{s}$ $+(5+k)\mathbf{m}$ $+\mathbf{S}_k + \mathbf{M}_k$

Fig. 6. Analysis of doubling for even embedding degree

Quantity	Formula	Modified Miller (Fig. 5, loop **2**)	Modified Miller (Fig. 5, loop **4**)
$(D\lambda)_{T,P}$	$(X_T - x_P Z_T^2)Z_T$	$\mathbf{s} + 2\mathbf{m}$	$\mathbf{s} + 2\mathbf{m}$
$(N\lambda)_{T,P}$	$Y_T - y_P Z_T^3$	$2\mathbf{m}$	$2\mathbf{m}$
$(N\ell)_{T,P}$	$(D\lambda)_{T,P}(y_Q - y_P) - (N\lambda)_{T,P}(x_Q - x_P)$		$2(k/2)\mathbf{m}$
X_{T+P}	$(N\lambda)_{T,P}^2 - X_T(X_T - x_P Z_T^2)^2$ $-x_P Z_T^2(X_T - x_P Z_T^2)^2$	$2\mathbf{s} + 2\mathbf{m}$	$2\mathbf{s} + 2\mathbf{m}$
Y_{T+P}	$-y_P Z_T^3(X_T(X_T - x_P Z_T^2)^2$ $-x_P Z_T^2(X_T - x_P Z_T^2)^2)$ $+(N\lambda)_{T,P}(x_P Z_T^2(X_T - x_P Z_T^2)^2$ $-X_{T+P})$	$2\mathbf{m}$	$2\mathbf{m}$
Z_{T+P}	$(D\lambda)_{T,P}$	0	0
$(N\ell')_{-T,-P}$	$\alpha_{Q,P}(D\lambda)_{T,P} + (N\lambda)_{T,P}$	$k\mathbf{m}$	
f	$\leftarrow f(N\ell')_{-T,-P}$	$\mathbf{M}_k$	
f	$\leftarrow f(N\ell)_{T,P}$		$\mathbf{M}_k$
TOTAL		$3\mathbf{s} + (8+k)\mathbf{m}$ $+\mathbf{M}_k$	$3\mathbf{s} + (8+k)\mathbf{m}$ $+\mathbf{M}_k$

Fig. 7. Cost analysis of addition for even embedding degree

- For $k = 18$, the curve is $E : y^2 = x^3 + 3$ and

  $r = 627710173538668076383578942320766641610235544446404691873 9$
  $q = 352868453926302204292551351775152292482424484549774231757$
  $0969033731372516464687041152677170740911665254402968138 9$

- For $k = 19$, the curve is $E : y^2 = x^3 + 2$ and

  $r = 627710173538668076383578942320766641610235544446403823192 7$
  $q = 328331730495825080256136908360300125975534969742697656797 5$
  $2651677037684673416288844142247623389652333826612345637$

For the computations, we used the NTL library [25] and implemented the algorithms without any optimization on an Intel(R) Core(TM)2 Duo CPU E8500 @ 3.16Ghz using Ubuntu Operating System 9.04. The computations of the Miller function (without any final exponentiation) were executed on 100 random inputs. The experimental average results are summarized in Figure 8.

k	Usual Miller (Fig. 1)	Our variant (Fig. 2)	Our variant with k even (Fig. 5)	Miller without denominators [4]
17	$0.0664s$	$0.0499s$	-	-
18	$0.0709s$	-	$0.0392s$	$0.0393s$
19	$0.0769s$	$0.0683s$	-	-

Fig. 8. Timings

6 Conclusion

In this paper we presented a variant of Miller's formula and algorithm. Generically, it is more efficient than the usual Miller algorithm as in Figure 1, calculation suggest that it can lead to a real improvement in cases where denominator elimination is not available. Consequently, we believe it will have applications in pairing-based cryptography using elliptic curves with embedding degree not being on the form $2^i 3^j$, for example when the optimal Ate or Twisted Ate pairing is used. Further work is needed to clarify such questions.

Acknowledgments

This work was supported by Project ANR-07-TCOM-013-04 PACE financed by the Agence National de Recherche (France). We would like to thank Andreas Enge for several helpful discussions and the anonymous reviewers for their numerous suggestions and remarks which have enables us to substantially improve the paper.

References

1. Christophe Arène, Tanja Lange, Michael Naehrig, and Christophe Ritzenthaler. Faster computation of the Tate pairing. Preprint, Cryptology ePrint Archive: Report 2009/155 `http://eprint.iacr.org/2009/155`, 2009.
2. Ramachandran Balasubramanian and Neal Koblitz. The improbability that an elliptic curve has subexponential discrete log problem under the Menezes–Okamoto–Vanstone algorithm. *J. Cryptology*, 11(2):141–145, 1998.
3. Paulo S. L. M. Barreto, Steven D. Galbraith, Colm O'Eigeartaigh, and Michael Scott. Efficient pairing computation on supersingular abelian varieties. *Des. Codes Cryptography*, 42(3):239–271, 2007.
4. Paulo S. L. M. Barreto, Ben Lynn, and Michael Scott. On the selection of pairing-friendly groups. In *Proc. of SAC 2003*, volume 3006 of *LNCS*, pages 17–25. Springer, 2003.
5. Paulo S. L. M. Barreto, Ben Lynn, and Michael Scott. Efficient implementation of pairing-based cryptosystems. *J. Cryptology*, 17(4):321–334, 2004.
6. Paulo S. L. M. Barreto and Michael Naehrig. Pairing-friendly elliptic curves of prime order. In *Proc. of SAC 2005*, volume 3897 of *LNCS*, pages 319–331. Springer, 2006.
7. Dan Bernstein and Tanja Lange. Explicit-formulas database, 2010. `http://www.hyperelliptic.org/EFD/`.
8. Ian F. Blake, V. Kumar Murty, and Guangwu Xu. Refinements of Miller's algorithm for computing the Weil/Tate pairing. *J. Algorithms*, 58(2):134–149, 2006.
9. Ian F. Blake, Gadiel Seroussi, and Nigel P. Smart. *Advances in Elliptic Curves Cryptography*, volume 317 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, Cambridge, 2005.

10. Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. *SIAM J. Comput.*, 32(3):586–615, 2003. Extended abstract in proc. of Crypto 2001.
11. Henri Cohen, Gerhard Frey, Roberto Avanzi, Christophe Doche, Tanja Lange, Kim Nguyen, and Frederik Vercauteren. *Handbook of Elliptic and Hyperelliptic Curve Cryptography*. Discrete mathematics and its applications. Chapman & Hall, Boca Raton, 2006.
12. Craig Costello, Huseyin Hisil, Colin Boyd, Juan Manuel Gonzalez Nieto, and Kenneth Koon-Ho Wong. Faster pairings on special Weierstrass curves. In *Proc. of Pairing 2009*, volume 5671 of *LNCS*, pages 89–101. Springer, 2009.
13. Craig Costello, Tanja Lange, and Michael Naehrig. Faster pairing computations on curves with high-degree twists. In *proc. of PKC 2010*, volume 6056, pages 224–242. Springer, 2010.
14. Nadia El Mrabet and Christophe Nègre. Finite field multiplication combining AMNS and DFT approach for pairing cryptography. In *Proc. of ACISP '09*, volume 5594 of *LNCS*, pages 422–436. Springer, 2009.
15. David Freeman, Michael Scott, and Edlyn Teske. A taxonomy of pairing-friendly elliptic curves. *J. Cryptology*, 23(2):224–280, 2010.
16. Florian Heß. Pairing lattices. In *Proc. of Pairing 2008*, volume 5209 of *LNCS*, pages 18–38. Springer, 2008.
17. Florian Heß, Nigel P. Smart, and Frederik Vercauteren. The Eta pairing revisited. *IEEE Transactions on Information Theory*, 52(10):4595–4602, 2006.
18. Sorina Ionica and Antoine Joux. Another approach to pairing computation in Edwards coordinates. In *Indocrypt 2008*, volume 5365 of *LNCS*, pages 400–413. Springer, 2008.
19. Sorina Ionica and Antoine Joux. Pairing the volcano. In *Proc. of ANTS-IX*, volume 6197 of *LNCS*, pages 201–218. Springer, 2010.
20. Antoine Joux. A one round protocol for tripartite Diffie-Hellman. *J. Cryptology*, 17(4):263–276, 2004. Extended abstract in proc. of ANTS 2000.
21. Neal Koblitz and Alfred Menezes. Pairing-based cryptography at high security levels. In *IMA Int. Conf. Cryptography and Coding*, volume 3796 of *LNCS*, pages 13–36. Springer, 2005.
22. Xibin Lin, Changan Zhao, Fangguo Zhang, and Yanming Wang. Computing the Ate pairing on elliptic curves with embedding degree $k = 9$. *IEICE Transactions*, 91-A(9):2387–2393, 2008.
23. Seiichi Matsuda, Naoki Kanayama, Florian Heß, and Eiji Okamoto. Optimised versions of the Ate and twisted Ate pairings. *IEICE Transactions*, 92-A(7):1660–1667, 2009.
24. Victor S. Miller. The Weil pairing, and its efficient calculation. *J. Cryptology*, 17(4):235–261, 2004.
25. Victor Shoup. NTL: a library for doing number theory, 2009. `http://www.shoup.net/ntl/`.
26. Frederik Vercauteren. Optimal pairings. *IEEE Transactions of Information Theory*, 56:455–461, 2009.