Speeding Up Pairing Computations on Genus 2 Hyperelliptic Curves with Efficiently Computable Automorphisms

Xinxin Fan[1,⋆], Guang Gong[1,⋆], and David Jao[2,⋆⋆]

[1] Department of Electrical and Computer Engineering
[2] Department of Combinatorics and Optimization
University of Waterloo
Waterloo, Ontario, N2L 3G1, CANADA
{x5fan@engmail,ggong@calliope,djao@math}.uwaterloo.ca

Abstract. Pairings on the Jacobians of (hyper-)elliptic curves have received considerable attention not only as a tool to attack curve based cryptosystems but also as a building block for constructing cryptographic schemes with new and novel properties. Motivated by the work of Scott, we investigate how to use efficiently computable automorphisms to speed up pairing computations on two families of non-supersingular genus 2 hyperelliptic curves over prime fields. Our findings lead to new variants of Miller's algorithm in which the length of the main loop can be up to 4 times shorter than that of the original Miller's algorithm in the best case. We also implement the calculation of the Tate pairing on both a supersingular and a non-supersingular genus 2 curve with the same embedding degree of $k = 4$. Combining the new algorithm with known optimization techniques, we show that pairing computations on non-supersingular genus 2 curves over prime fields use up to 55.8% fewer field operations and run about 10% faster than supersingular genus 2 curves for the same security level.

Keywords: Genus 2 non-supersingular hyperelliptic curves, Tate pairing, Miller's algorithm, Automorphism, Efficient implementation.

1 Introduction

Pairing based cryptography is a relatively new area of cryptography centered around particular functions with interesting properties. Initially, bilinear pairings were introduced to cryptography for attacking instances of the discrete logarithm problem (DLP) on elliptic curves and hyperelliptic curves [14,28]. With the advent of non-interactive key distribution [33], tripartite key exchange [24], and identity based encryption [5], the design of pairing based cryptographic protocols has received a lot of attention from the research community. The implementation of pairing based protocols requires the efficient computation of

⋆ Supported by NSERC Strategic Project Grants (SPG).
⋆⋆ Partially supported by NSERC.

S.D. Galbraith and K.G. Paterson (Eds.): Pairing 2008, LNCS 5209, pp. 243–264, 2008.

pairings. To date, the Weil and Tate pairings and their variants such as the Eta and Ate pairings on Jacobians of (hyper-)elliptic curves are the only efficient instantiations of cryptographically useful bilinear maps.

There has been a lot of work on efficient implementation of pairings on elliptic curves, and many important techniques have been proposed to accelerate the computation of the Tate pairing and its variants on elliptic curves [2,3,4,22]. Furthermore, the subject of pairing computations on hyperelliptic curves is also receiving an increasing amount of attention. Choie and Lee [6] investigated the implementation of the Tate pairing on supersingular genus 2 hyperelliptic curves over prime fields. Later on, hÉigeartaigh and Scott [21] improved the implementation of [6] significantly by using a new variant of Miller's algorithm combined with various optimization techniques. Duursma and Lee [10] presented a closed formula for the Tate pairing computation on a very special family of supersingular hyperelliptic curves. Barreto *et. al.* [2] generalized the results of Duursma and Lee and proposed the Eta pairing approach for efficiently computing the Tate pairing on supersingular genus 2 curves over binary fields. In particular, their algorithm leads to the fastest pairing implementation in the literature. In [27], Lee *et. al.* considered the Eta pairing computation on general divisors on supersingular genus 3 hyperelliptic curves of the form of $y^2 = x^7 - x \pm 1$. Recently, the Ate pairing, originally defined for elliptic curves, has been generalized to hyperelliptic curves [18] as well. Although the Eta and Ate pairings hold the record for speed at the present time, we will focus on the Tate pairing in this paper. The main reason is that the Tate pairing is uniformly available across a wide range of hyperelliptic curves and subgroups, whereas the Eta pairing is only defined for supersingular curves and the Ate pairing incurs a huge performance penalty in the context of ordinary genus 2 curves [18, Table 6].

Motivated by previous work in [34,38,39], we consider pairing computations on two families of non-supersingular genus 2 hyperelliptic curves over prime fields. We first explicitly give efficiently computable automorphisms (also isogenies) and the dual isogenies on the divisor class group of these curves. We then exploit the automorphism in lieu of the order of the torsion group to construct the rational functions required in Miller's algorithm, and shorten the length of the main loop in Miller's algorithm as a result. Based on the new construction of the rational functions, we propose new variants of Miller's algorithm for computing the Tate pairing on certain non-supersingular genus 2 curves over prime fields. In the best case, the length of the loop in our new variant can be up to 4 times shorter than that of the original Miller's algorithm. Furthermore, we generate a non-supersingular pairing-friendly genus 2 curve with embedding degree 4 and compare the performance of our new algorithm with that of the variant proposed by hÉigeartaigh and Scott [21] for supersingular genus 2 curves. Theoretical analysis shows that our new algorithm uses 55.8% fewer field operations than that of [21] for the same security level. However, the size of the field where the non-supersingular curve is defined is 1.285 times larger than that of the field used for supersingular curves, which somewhat offsets these gains. Nevertheless, our experimental results show that using the non-supersingular genus 2 curve one can still obtain a 10% performance improvement over the supersingular curve.

The rest of this paper is organized as follows. Section 2 gives a short introduction to the Tate pairing on hyperelliptic curves and Miller's algorithm for computing the pairing. In Section 3 we recall supersingular genus 2 curves over prime fields which have been used for pairing computations, and introduce two families of non-supersingular genus 2 curves with efficiently computable automorphisms. In Section 4 we prove the main results of our contribution and propose new variants of Miller's algorithm. Section 5 details the various techniques for efficiently implementing the Tate pairing on a non-supersingular genus 2 curve with embedding degree 4, analyzes the computational cost of our new algorithm and gives experimental results. Finally, Section 6 concludes this paper.

2 Mathematical Background

In this section, we present a brief introduction to the definition of the Tate pairing on hyperelliptic curves and also review Miller's algorithm for computing pairings. For more details, the reader is referred to [1].

2.1 Tate Pairing on Hyperelliptic Curves

Let $\mathbb{F}_q$ be a finite field with q elements, and $\overline{\mathbb{F}}_q$ be its algebraic closure. Let C be a hyperelliptic curve of genus g over $\mathbb{F}_q$, and let $\mathcal{J}_C$ denote the degree zero divisor class group of C. We say that a subgroup of the divisor class group $\mathcal{J}_C(\mathbb{F}_q)$ has *embedding degree* k if the order n of the subgroup divides $q^k - 1$, but does not divide $q^i - 1$ for any $0 < i < k$. For our purpose, n should be a (large) prime with $n \mid \#\mathcal{J}_C(\mathbb{F}_q)$ and $\gcd(n, q) = 1$. Let $\mathcal{J}_C(\mathbb{F}_{q^k})[n]$ be the n-torsion group and $\mathcal{J}_C(\mathbb{F}_{q^k})/n\mathcal{J}_C(\mathbb{F}_{q^k})$ be the quotient group. Then the Tate pairing is a well defined, non-degenerate, bilinear map [14]:

$$\langle \cdot, \cdot \rangle_n : \mathcal{J}_C(\mathbb{F}_{q^k})[n] \times \mathcal{J}_C(\mathbb{F}_{q^k})/n\mathcal{J}_C(\mathbb{F}_{q^k}) \rightarrow \mathbb{F}_{q^k}^*/(\mathbb{F}_{q^k}^*)^n,$$

defined as follows: let $D_1 \in \mathcal{J}_C(\mathbb{F}_{q^k})[n]$, with $\mathrm{div}(f_{n,D_1}) = nD_1$ for some rational function $f_{n,D_1} \in \mathbb{F}_{q^k}(C)^*$. Let $D_2 \in \mathcal{J}_C(\mathbb{F}_{q^k})/n\mathcal{J}_C(\mathbb{F}_{q^k})$ with $\mathrm{supp}(D_1) \cap \mathrm{supp}(D_2) = \emptyset$ (to ensure a non-trivial pairing value). The Tate pairing of two divisor classes $\overline{D}_1$ and $\overline{D}_2$ is then defined by

$$\langle \overline{D}_1, \overline{D}_2 \rangle_n = f_{n,D_1}(D_2) = \prod_{P \in C(\overline{\mathbb{F}}_q)} f_{n,D_1}(P)^{\mathrm{ord}_P(D_2)}.$$

Note that the Tate pairing as detailed above is only defined up to n-th powers. One can show that if the function f_{n,D_1} is properly normalized, we only need to evaluate the rational function f_{n,D_1} at the effective part of the reduced divisor D_2 in order to compute the Tate pairing [3,18].

In practice, the fact that the Tate pairing is only defined up to n-th power is usually undesirable, and many pairing-based protocols require a unique pairing value. Hence one defines the *reduced* pairing as

$$\langle \overline{D}_1, \overline{D}_2 \rangle_n^{(q^k-1)/n} = f_{n,D_1}(D_2)^{(q^k-1)/n} \in \mu_n \subset \mathbb{F}_{q^k}^*,$$

where $\mu_n = \{u \in \mathbb{F}_{q^k}^* \mid u^n = 1\}$ is the group of n-th roots of unity. In the rest of this paper we will refer to the extra powering required to compute the reduced pairing as the *final exponentiation*. One of the important properties of the reduced pairing we will use in this paper is that for any positive integer N satisfying $n \mid N$ and $N \mid q^k - 1$ we have

$$\langle D_1, D_2 \rangle_n^{(q^k-1)/n} = \langle D_1, D_2 \rangle_N^{(q^k-1)/N}. \tag{1}$$

2.2 Miller's Algorithm

The main task involved in the computation of the Tate pairing $\langle \overline{D}_1, \overline{D}_2 \rangle_n$ is to construct a rational function f_{n,D_1} such that $\text{div}(f_{n,D_1}) = nD_1$. In [29] (see also [30]), Miller described a polynomial time algorithm, known universally as Miller's algorithm, to construct the function f_{n,D_1} and compute the Weil pairing on elliptic curves. However, the algorithm can be easily adapted to compute the Tate pairing on hyperelliptic curves.

Let $G_{iD_1,jD_1} \in \mathbb{F}_{q^k}(C)^*$ be a rational function with $\text{div}(G_{iD_1,jD_1}) = iD_1 + jD_1 - (iD_1 \oplus jD_1)$ where $\oplus$ is the group law on $\mathcal{J}_C$ and $(iD_1 \oplus jD_1)$ is reduced. Miller's algorithm constructs the rational function f_{n,D_1} based on the following iterative formula:

$$f_{i+j,D_1} = f_{i,D_1} f_{j,D_1} G_{iD_1,jD_1}.$$

The following Algorithm 1 shows the basic version of Miller's algorithm for computing the reduced Tate pairing on hyperelliptic curves according to the above iterative relation. A more detailed version of Miller's algorithm for hyperelliptic curves can be found in [18].

Choie and Lee [6] described how to explicitly find the rational function $G(x, y)$ in the Algorithm 1 for the case of genus 2 hyperelliptic curves. Their results can be summarized as follows: Let $D_1 = [u_1, v_1]$ and $D_2 = [u_2, v_2]$ be the two reduced divisors in $\mathcal{J}_C(\mathbb{F}_{q^k})$ that are being added. In the composition stage of Cantor's algorithm, we compute the polynomial $\delta(x)$ which is the greatest common divisor of u_1, u_2 and $v_1 + v_2 + h$ and a divisor $D_3' = [u_3', v_3']$, which is in the same divisor class as $D_3 = [u_3, v_3] = \overline{D}_1 + \overline{D}_2$. At this point, two cases may occur:

1. If the divisor D_3' is already reduced following the composition stage, then the rational function $G(x, y) = \delta(x)$.
2. If this is not the case, then the rational function $G(x, y) = \delta(x) \cdot \frac{y - v_3'(x)}{u_3(x)}$.

In the most frequent cases[1] $\delta = 1$ and thus $G(x, y) = \frac{y - v_3'(x)}{u_3(x)}$.

[1] For addition, the inputs are two co-prime polynomials of degree 2, and for doubling the input is a square free polynomial of degree 2.

Algorithm 1. Miller's Algorithm for Hyperelliptic Curves (basic version)

IN: $\overline{D}_1 \in \mathcal{J}_C(\mathbb{F}_{q^k})[n], \overline{D}_2 \in \mathcal{J}_C(\mathbb{F}_{q^k})$, represented by D_1 and D_2
with $\text{supp}(D_1) \cap \text{supp}(D_2) = \emptyset$
OUT: $\langle D_1, D_2 \rangle_n^{(q^k-1)/n}$
1. $f \leftarrow 1, T \leftarrow D_1$
2. for $i \leftarrow \lfloor \log_2(n) \rfloor - 1$ **downto 0 do**
3. $\quad \triangleright$ Compute T' and $G_{T,T}(x, y)$ such that $T' = 2T - \text{div}(G_{T,T})$
4. $\quad\quad f \leftarrow f^2 \cdot G_{T,T}(D_2), \overline{T} \leftarrow [2]\overline{T}$
5. if $n_i = 1$ **then**
6. $\quad \triangleright$ Compute T' and $G_{T,D_1}(x, y)$ such that $T' = T + D_1 - \text{div}(G_{T,D_1})$
7. $\quad\quad f \leftarrow f \cdot G_{T,D_1}(D_2), \overline{T} \leftarrow \overline{T} \oplus \overline{D}_1$
8. Return $f^{(q^k-1)/n}$

3 Supersingular Curves and Non-supersingular Curves

In this section, we first recall the supersingular genus 2 curves over $\mathbb{F}_p$ which have been used to implement the Tate pairing. Then, by making a small change to the definition of these curves, we produce two families of non-supersingular genus 2 curves over $\mathbb{F}_p$ with efficiently computable automorphisms which provide potential advantages for pairing computations.

3.1 Supersingular Genus 2 Curves over $\mathbb{F}_p$

Theoretically, supersingular genus 2 hyperelliptic curves exist over $\mathbb{F}_p$ with an embedding degree of $k = 6$ [32]. However, only supersingular genus 2 curves with an embedding degree of $k = 4$ are known to the cryptographic community at the present time [7]. In [6,21], the authors investigated the efficient implementation of the Tate pairing on supersingular genus 2 curves with embedding degree 4. The curve used in their implementation is defined by the following equation:

$$C_1 : y^2 = x^5 + a, \;\; a \in \mathbb{F}_p^* \;\text{ and }\; p \equiv 2, 3 \pmod 5.$$

On these supersingular curves a *modified pairing* $\langle D_1, \psi(D_1) \rangle_n^{(p^k-1)/n}$ is computed, where the map $\psi_1(\cdot)$ is a *distortion map* that maps elements in $C_1(\mathbb{F}_p)$ to $C_1(\mathbb{F}_{p^4})$. The distortion map ψ_1 is given by $\psi_1(x, y) = (\xi_5 x, y)$, where ξ_5 is a primitive 5-th root of unity in $\mathbb{F}_{p^4}$. We also note that another family of supersingular genus 2 curves over $\mathbb{F}_p$ with embedding degree 4 [7] is also suitable for implementing pairings. Such curves are given by an equation of the form

$$C_2 : y^2 = x^5 + ax, \;\; a \in \mathbb{F}_p^* \cap \overline{QR}_p \;\text{ and }\; p \equiv 5 \pmod 8,$$

where $\overline{QR}_p$ denotes the set of quadratic non-residues modulo p. The distortion map for the curve C_2 is defined by $\psi_2(x, y) = (\xi_8^2 x, \xi_8 y)$, where ξ_8 is a primitive 8-th root of unity in $\mathbb{F}_{p^4}$.

3.2 Non-supersingular Genus 2 Curves over $\mathbb{F}_p$

Motivated by the work in [34,38,39], we consider now the following two families of non-supersingular genus 2 hyperelliptic curves over $\mathbb{F}_p$:

$$C_1' : y^2 = x^5 + a, \quad a \in \mathbb{F}_p^* \quad \text{and} \quad p \equiv 1 \pmod 5,$$
$$C_2' : y^2 = x^5 + ax, \quad a \in \mathbb{F}_p^* \quad \text{and} \quad p \equiv 1 \pmod 8.$$

Curves of this form exist which are pairing friendly (see Section 4). Note that the only difference between the curves C_i and C_i' $(i = 1, 2)$ is the congruence condition applied to the characteristic p. Although distortion maps do not exist on these non-supersingular curves, both C_1' and C_2' have efficiently-computable *endomorphisms*. In fact, these endomorphisms also induce efficient *automorphisms* on the divisor class groups of C_1' and C_2', which have been used to accelerate the scalar multiplication for hyperelliptic curve cryptosystems [31] and to attack the discrete log problems on the Jacobians [9,17]. In the next section, we will show how to speed up the computation of the Tate pairing using these efficiently computable automorphisms on the curves C_1' and C_2'. We first recall some basic facts which will be used in this work.

For the curve C_1', the morphism ψ_1 defined by $P = (x, y) \mapsto \psi_1(P) = (\xi_5 x, y)$ (see Section 3.1 and notice $\xi_5 \in \mathbb{F}_p$ now) induces an efficient non-trivial automorphism of order 5 on the divisor class group $\mathcal{J}_{C_1'}(\mathbb{F}_p)$ [31]. The formulae for ψ_1 on the Jacobian are given by

$$\begin{aligned} \psi_1 : [x^2 + u_1 x + u_0, v_1 x + v_0] &\mapsto [x^2 + \xi_5 u_1 x + \xi_5^2 u_0, \xi_5^{-1} v_1 x + v_0] \\ [x + u_0, v_0] &\mapsto [x + \xi_5 u_0, v_0] \\ \mathcal{O} &\mapsto \mathcal{O}. \end{aligned}$$

The characteristic polynomial of ψ_1 is given by $P(t) = t^4 + t^3 + t^2 + t + 1$. Since the automorphism ψ_1 is also an *isogeny*, we can construct its *dual isogeny* as follows:

$$\begin{aligned} \widehat{\psi}_1 : [x^2 + u_1 x + u_0, v_1 x + v_0] &\mapsto [x^2 + \xi_5^{-1} u_1 x + \xi_5^{-2} u_0, \xi_5 v_1 x + v_0] \\ [x + u_0, v_0] &\mapsto [x + \xi_5^{-1} u_0, v_0] \\ \mathcal{O} &\mapsto \mathcal{O}. \end{aligned}$$

Note that $\psi_1 \circ \widehat{\psi}_1 = [1]$ and $\#\,\mathrm{Ker}\ \psi_1 = \deg[1] = 1$, and $\widehat{\psi}_1$ is also a non-trivial automorphism on the curve C_1'.

Let $D \in \mathcal{J}_{C_1'}(\mathbb{F}_p)$ be a reduced divisor of a large prime order n. From [31], we know that the automorphisms ψ_1 and $\widehat{\psi}_1$ act respectively as multiplication maps by $[\lambda_1]$ and $[\widehat{\lambda}_1]$ on the subgroup $\langle D \rangle$ of $\mathcal{J}_{C_1'}(\mathbb{F}_p)$, where λ_1 and $\widehat{\lambda}_1$ are the two roots of the equation $t^4 + t^3 + t^2 + t + 1 \equiv 0 \pmod n$. Furthermore, it is easily seen that $[\lambda_1]D = \psi_1(D)$ can be obtained with only three or one field multiplications in $\mathbb{F}_p$ for a *general divisor* and a *degenerate divisor*, respectively. In the genus 2 context, a general divisor has two finite points in the support, whereas a degenerate divisor has only a single finite point in its support.

Similarly, for the curve C_2', the morphism ψ_2 defined by $P = (x, y) \mapsto \psi_2(P) = (\xi_8^2 x, \xi_8 y)$ (see Section 3.1 and notice $\xi_8 \in \mathbb{F}_p$ now) induces an efficient non-trivial automorphism of order 8 on the divisor class group $\mathcal{J}_C(\mathbb{F}_p)$ as follows [31].

$$\begin{aligned}\psi_2 : [x^2 + u_1 x + u_0, v_1 x + v_0] &\mapsto [x^2 + \xi_8^2 u_1 x + \xi_8^4 u_0, \xi_8^{-1} v_1 x + \xi_8 v_0] \\ [x + u_0, v_0] &\mapsto [x + \xi_8^2 u_0, \xi_8 v_0] \\ \mathcal{O} &\mapsto \mathcal{O}.\end{aligned}$$

The characteristic polynomial of ψ_2 is given by $P(t) = t^4 + 1$ and the dual isogeny of ψ_2 is defined as follows

$$\begin{aligned}\widehat{\psi}_2 : [x^2 + u_1 x + u_0, v_1 x + v_0] &\mapsto [x^2 + \xi_8^{-2} u_1 x + \xi_8^4 u_0, \xi_8 v_1 x + \xi_8^{-1} v_0] \\ [x + u_0, v_0] &\mapsto [x + \xi_8^{-2} u_0, \xi_8^{-1} v_0] \\ \mathcal{O} &\mapsto \mathcal{O}.\end{aligned}$$

It is not difficult to show that $\psi_2 \circ \widehat{\psi}_2 = [1]$ and $\#\,\mathrm{Ker}\ \psi_2 = \deg[1] = 1$, and $\widehat{\psi}_2$ is also a non-trivial automorphism on the curve C_2'. Let $D \in \mathcal{J}_{C_2'}(\mathbb{F}_p)$ be a reduced divisor of a large prime order n. Then the automorphism ψ_2 acts as a multiplication map by λ_2 on the subgroup $\langle D \rangle$ of $\mathcal{J}_{C_2'}(\mathbb{F}_p)$, where λ_2 is a root of the equation $t^4 + 1 \equiv 0 \pmod{n}$. Moreover, $[\lambda_2]D = \psi_2(D)$ can be computed with four or two field multiplications in $\mathbb{F}_p$ for a general divisor and a degenerate divisor, respectively.

4 Efficient Pairings on Non-supersingular Genus 2 Curves

In this section we investigate efficient algorithms for computing the Tate pairing on the two families of genus 2 hyperelliptic curves C_1' and C_2' defined in Section 3.2. We show how to use the efficiently-computable automorphisms on these curves to shorten the length of the loop in Miller's algorithm. As a result, we propose new variants of Miller's algorithm for certain non-supersingular genus 2 curves over large prime fields.

4.1 Pairing Computation with Efficient Automorphisms

In this subsection, we present the main results of this paper in the following theorems and show their correctness. The pairing computation on the curve C_1' is first examined.

Theorem 1. *Let C_1' be a non-supersingular genus 2 hyperelliptic curve over $\mathbb{F}_p$ as above, with embedding degree $k > 1$ and automorphisms ψ_1 and $\widehat{\psi}_1$ defined as above. Let $D_1 = [u_1(x), v_1(x)] \in \mathcal{J}_{C_1'}(\mathbb{F}_p)$ be a reduced divisor of prime order n, where $n^2 \nmid \#\mathcal{J}_{C_1'}(\mathbb{F}_p)$. Let $[\lambda_1]$ act as the multiplication map on the subgroup $\langle D_1 \rangle$ defined as above such that $[\lambda_1]D_1 = \psi_1(D_1)$. Suppose $m \in \mathbb{Z}$ is such that*

$mn = \lambda_1^4+\lambda_1^3+\lambda_1^2+\lambda_1+1$. *Let rational functions* $\frac{c_1(x,y)}{d_1(x,y)}, \frac{c_2(x,y)}{d_2(x,y)}, \frac{c_3(x,y)}{d_3(x,y)} \in \mathbb{F}_p(C_1')^*$ *respectively satisfy the following three relations:*

$$[\lambda_1]D_1 + [\lambda_1^2]D_1 - \left([\lambda_1]D_1 \oplus [\lambda_1^2]D_1\right) = \mathrm{div}\left(\frac{c_1(x,y)}{d_1(x,y)}\right),$$

$$\left[\lambda_1^3\right] D_1 + [\lambda_1^4]D_1 - \left([\lambda_1^3]D_1 \oplus [\lambda_1^4]D_1\right) = \mathrm{div}\left(\frac{c_2(x,y)}{d_2(x,y)}\right),$$

$$\left[\lambda_1 + \lambda_1^2\right] D_1 + [\lambda_1^3 + \lambda_1^4]D_1 - \left([\lambda_1 + \lambda_1^2]D_1 \oplus [\lambda_1^3 + \lambda_1^4]D_1\right) = \mathrm{div}\left(\frac{c_3(x,y)}{d_3(x,y)}\right).$$

Let $g(x,y) = \frac{c_1(x,y)\cdot c_2(x,y)\cdot c_3(x,y)}{d_1(x,y)\cdot d_2(x,y)}$. *Then for* $D_2 \in \mathcal{J}_{C_1'}(\mathbb{F}_{p^k})$*, we have*

$$\langle D_1, D_2\rangle_n^{\frac{m(p^k-1)}{n}} = \Big[f_{\lambda_1,D_1}^{\lambda_1^3+\lambda_1^2+\lambda_1+1}(D_2)\cdot f_{\lambda_1,D_1}^{\lambda_1^2+\lambda_1+1}\left(\widehat{\psi}_1(D_2)\right)\cdot f_{\lambda_1,D_1}^{\lambda_1+1}\left(\widehat{\psi}_1^2(D_2)\right)\cdot$$
$$f_{\lambda_1,D_1}\left(\psi_1^2(D_2)\right)\cdot g(D_2)\Big]^{\frac{p^k-1}{n}}.$$

Note that the condition that λ_1 satisfies $\lambda_1^4+\lambda_1^3+\lambda_1^2+\lambda_1+1 \equiv 0 \pmod n$ guarantees the existence of the integer m. Moreover, the pairing will be non-degenerate if $n \nmid m$ and $\mathrm{supp}(D_1)\cap\mathrm{supp}(D_2) = \emptyset$. We split the proof of the Theorem 1 into the following lemmas.

Lemma 1. *With notation as above, we have*

$$\langle D_1, D_2\rangle_n^{\frac{m(p^k-1)}{n}} = \left(f_{\lambda_1^4+\lambda_1^3+\lambda_1^2+\lambda_1,D_1}(D_2)\cdot u_1(D_2)\right)^{\frac{p^k-1}{n}}.$$

Proof. From the important property of the reduced pairing (see equation (1)), we have

$$\langle D_1, D_2\rangle_n^{\frac{m(p^k-1)}{n}} = \langle D_1, D_2\rangle_{mn}^{\frac{p^k-1}{n}} = f_{mn,D_1}(D_2)^{\frac{p^k-1}{n}}.$$

From the condition that $mn = \lambda_1^4+\lambda_1^3+\lambda_1^2+\lambda_1+1$, we get

$$\langle D_1, D_2\rangle_n^{\frac{m(p^k-1)}{n}} = f_{mn,D_1}(D_2)^{\frac{p^k-1}{n}} = f_{\lambda_1^4+\lambda_1^3+\lambda_1^2+\lambda_1+1,D_1}(D_2)^{\frac{p^k-1}{n}}.$$

Since $[\lambda_1^4+\lambda_1^3+\lambda_1^2+\lambda_1]D_1 = -D_1$, we obtain the following relation

$$D_1 + [\lambda_1+\lambda_1^2+\lambda_1^3+\lambda_1^4]D_1 = D_1 + (-D_1) = \mathrm{div}(u_1(x)).$$

Therefore, we have

$$\begin{aligned}&\mathrm{div}\left(f_{\lambda_1^4+\lambda_1^3+\lambda_1^2+\lambda_1+1,D_1}\right) = (\lambda_1^4+\lambda_1^3+\lambda_1^2+\lambda_1)D_1 + D_1\\&= \mathrm{div}\left(f_{\lambda_1^4+\lambda_1^3+\lambda_1^2+\lambda_1,D_1}\right) + D_1 + [\lambda_1+\lambda_1^2+\lambda_1^3+\lambda_1^4]D_1\\&= \mathrm{div}\left(f_{\lambda_1^4+\lambda_1^3+\lambda_1^2+\lambda_1,D_1}\cdot u_1(x)\right),\end{aligned}$$

which proves the result. □

The next lemma shows the relation between $\operatorname{div}\left(f_{\lambda_1^4+\lambda_1^3+\lambda_1^2+\lambda_1,D_1} \cdot u_1(x)\right)$ and the divisors $\operatorname{div}\left(f_{\lambda_1,[\lambda_1^i]D_1}\right)$ for $i = 0, 1, 2,$ and 3.

Lemma 2. *With notation as above, we have*

$$\operatorname{div}\left(f_{\lambda_1^4+\lambda_1^3+\lambda_1^2+\lambda_1,D_1} \cdot u_1(x)\right) = \operatorname{div}\left(f_{\lambda_1,D_1}^{\lambda_1^3+\lambda_1^2+\lambda_1+1} \cdot f_{\lambda_1,[\lambda_1]D_1}^{\lambda_1^2+\lambda_1+1} \cdot f_{\lambda_1,[\lambda_1^2]D_1}^{\lambda_1+1} \cdot f_{\lambda_1,[\lambda_1^3]D_1} \cdot g(x,y)\right).$$

Proof. We first note the following relation

$$\begin{aligned}\operatorname{div}\left(f_{\lambda_1^4+\lambda_1^3+\lambda_1^2+\lambda_1,D_1}\right) &= (\lambda_1^4+\lambda_1^3+\lambda_1^2+\lambda_1)D_1 - [\lambda_1^4+\lambda_1^3+\lambda_1^2+\lambda_1]D_1 \\ &= \operatorname{div}\left(f_{\lambda_1^4+\lambda_1^3,D_1}\right) + \operatorname{div}\left(f_{\lambda_1^2+\lambda_1,D_1}\right) + [\lambda_1+\lambda_1^2]D_1 + \\ &\quad [\lambda_1^3+\lambda_1^4]D_1 - \left([\lambda_1+\lambda_1^2]D_1 \oplus [\lambda_1^3+\lambda_1^4]D_1\right) \\ &= \operatorname{div}\left(f_{\lambda_1^4+\lambda_1^3,D_1}\right) + \operatorname{div}\left(f_{\lambda_1^2+\lambda_1,D_1}\right) + \operatorname{div}\left(\frac{c_3(x,y)}{d_3(x,y)}\right) \\ &= \operatorname{div}\left(f_{\lambda_1^4+\lambda_1^3,D_1} \cdot f_{\lambda_1^2+\lambda_1,D_1} \cdot \frac{c_3(x,y)}{d_3(x,y)}\right)\end{aligned}$$

Since $[\lambda_1^4+\lambda_1^3+\lambda_1^2+\lambda_1]D_1 = -D_1$, we get $d_3(x,y) = u_1(x)$. Therefore, we have

$$\operatorname{div}\left(f_{\lambda_1^4+\lambda_1^3+\lambda_1^2+\lambda_1,D_1} \cdot u_1(x)\right) = \operatorname{div}\left(f_{\lambda_1^4+\lambda_1^3,D_1} \cdot f_{\lambda_1^2+\lambda_1,D_1} \cdot c_3(x,y)\right). \quad (2)$$

Similarly, we can obtain the following two equalities

$$\begin{aligned}&\operatorname{div}\left(f_{\lambda_1^4+\lambda_1^3,D_1}\right) = (\lambda_1^4+\lambda_1^3)D_1 - [\lambda_1^4+\lambda_1^3]D_1 \\ &= \operatorname{div}\left(f_{\lambda_1^4,D_1}\right) + \operatorname{div}\left(f_{\lambda_1^3,D_1}\right) + [\lambda_1^4]D_1 + [\lambda_1^3]D_1 - \left([\lambda_1^3]D_1 \oplus [\lambda_1^4]D_1\right) \\ &= \operatorname{div}\left(f_{\lambda_1^4,D_1}\right) + \operatorname{div}\left(f_{\lambda_1^3,D_1}\right) + \operatorname{div}\left(\frac{c_2(x,y)}{d_2(x,y)}\right) \\ &= \operatorname{div}\left(f_{\lambda_1^4,D_1} \cdot f_{\lambda_1^3,D_1} \cdot \frac{c_2(x,y)}{d_2(x,y)}\right)\end{aligned}$$

and

$$\begin{aligned}&\operatorname{div}\left(f_{\lambda_1^2+\lambda_1,D_1}\right) = (\lambda_1^2+\lambda_1)D_1 - [\lambda_1^2+\lambda_1]D_1 \\ &= \operatorname{div}\left(f_{\lambda_1^2,D_1}\right) + \operatorname{div}\left(f_{\lambda_1,D_1}\right) + [\lambda_1^2]D_1 + [\lambda_1]D_1 - \left([\lambda_1]D_1 \oplus [\lambda_1^2]D_1\right) \\ &= \operatorname{div}\left(f_{\lambda_1^2,D_1}\right) + \operatorname{div}\left(f_{\lambda_1,D_1}\right) + \operatorname{div}\left(\frac{c_1(x,y)}{d_1(x,y)}\right) \\ &= \operatorname{div}\left(f_{\lambda_1^2,D_1} \cdot f_{\lambda_1,D_1} \cdot \frac{c_1(x,y)}{d_1(x,y)}\right)\end{aligned}$$

Some easy calculations (see Lemma 2 in [2]) give us

$$\operatorname{div}\left(f_{\lambda_1^4,D_1}\right) = \operatorname{div}\left(f^{\lambda_1^3}_{\lambda_1,D_1} \cdot f^{\lambda_1^2}_{\lambda_1,[\lambda_1]D_1} \cdot f^{\lambda_1}_{\lambda_1,[\lambda_1^2]D_1} \cdot f_{\lambda_1,[\lambda_1^3]D_1}\right) \quad (3)$$

$$\operatorname{div}\left(f_{\lambda_1^3,D_1}\right) = \operatorname{div}\left(f^{\lambda_1^2}_{\lambda_1,D_1} \cdot f^{\lambda_1}_{\lambda_1,[\lambda_1]D_1} \cdot f_{\lambda_1,[\lambda_1^2]D_1}\right) \quad (4)$$

$$\operatorname{div}\left(f_{\lambda_1^2,D_1}\right) = \operatorname{div}\left(f^{\lambda_1}_{\lambda_1,D_1} \cdot f_{\lambda_1,[\lambda_1]D_1}\right) \quad (5)$$

Combining the equations (2)–(7) proves the result. □

The following lemma shows how to evaluate functions $f_{\lambda_1,[\lambda_1^i]D_1}$ $(i = 1, 2, 3)$ at the image divisor D_2 by using the function f_{λ_1,D_1}.

Lemma 3. *With notation as above, we have (up to a scalar multiple in* $\mathbb{F}_p^*$*)*

$$f_{\lambda_1,[\lambda_1]D_1}(D_2) = f_{\lambda_1,D_1}(\widehat{\psi}_1(D_2)),$$
$$f_{\lambda_1,[\lambda_1^2]D_1}(D_2) = f_{\lambda_1,D_1}(\widehat{\psi}_1^2(D_2)),$$
$$f_{\lambda_1,[\lambda_1^3]D_1}(D_2) = f_{\lambda_1,D_1}(\psi_1^2(D_2)).$$

Proof. Using the pullback property (see Silverman [36] p. 33) and following the same proof as the Lemma 1 in [2], we obtain (up to a scalar multiple in $\mathbb{F}_p^*$)

$$f_{\lambda_1,[\lambda_1]D_1} \circ \psi_1 = f_{\lambda_1,D_1},$$
$$f_{\lambda_1,[\lambda_1^2]D_1} \circ \psi_1^2 = f_{\lambda_1,D_1},$$
$$f_{\lambda_1,[\lambda_1^3]D_1} \circ \psi_1^3 = f_{\lambda_1,D_1}.$$

Using the relations between the isogeny ψ_1 and its dual isogeny $\widehat{\psi}_1$ (see Section 3.2), we have

$$f_{\lambda_1,[\lambda_1]D_1} \circ \psi_1 \circ \widehat{\psi}_1 = f_{\lambda_1,[\lambda_1]D_1} = f_{\lambda_1,D_1} \circ \widehat{\psi}_1,$$
$$f_{\lambda_1,[\lambda_1^2]D_1} \circ \psi_1^2 \circ \widehat{\psi}_1^2 = f_{\lambda_1,[\lambda_1^2]D_1} = f_{\lambda_1,D_1} \circ \widehat{\psi}_1^2,$$
$$f_{\lambda_1,[\lambda_1^3]D_1} \circ \psi_1^3 \circ \widehat{\psi}_1^3 = f_{\lambda_1,[\lambda_1^3]D_1} = f_{\lambda_1,D_1} \circ \widehat{\psi}_1^3 = f_{\lambda_1,D_1} \circ \psi_1^2,$$

which proves the results. □

With the above three lemmas, we can now prove the statement of Theorem 1 as follows:

Proof (of Theorem 1). For $D_1 \in \mathcal{J}_{C_1'}(\mathbb{F}_p)[n]$ and $D_2 \in \mathcal{J}_{C_1'}(\mathbb{F}_{p^k})$, Lemma 3 shows that up to a scalar multiple in $\mathbb{F}_p^*$ we have

$$f_{\lambda_1,[\lambda_1]D_1}(D_2) = f_{\lambda_1,D_1}(\widehat{\psi}_1(D_2)),$$
$$f_{\lambda_1,[\lambda_1^2]D_1}(D_2) = f_{\lambda_1,D_1}(\widehat{\psi}_1^2(D_2)),$$
$$f_{\lambda_1,[\lambda_1^3]D_1}(D_2) = f_{\lambda_1,D_1}(\psi_1^2(D_2)).$$

Now, substituting the above three equalities into Lemma 2 implies that

$$f_{\lambda_1^4+\lambda_1^3+\lambda_1^2+\lambda_1,D_1}(D_2)\cdot u_1(D_2) = f_{\lambda_1,D_1}^{\lambda_1^3+\lambda_1^2+\lambda_1+1}(D_2)\cdot f_{\lambda_1,D_1}^{\lambda_1^2+\lambda_1+1}\left(\widehat{\psi}_1(D_2)\right)\cdot f_{\lambda_1,D_1}^{\lambda_1+1}\left(\widehat{\psi}_1^2(D_2)\right)\cdot f_{\lambda_1,D_1}\left(\psi_1^2(D_2)\right)\cdot g(D_2).$$

Finally, substituting the above equation into Lemma 1 gives the result of Theorem 1. □

Next, we consider how to use the efficiently-computable automorphism ψ_2 to accelerate the computation of the Tate pairing on the curve C_2'. The following Theorem 2 describes our result.

Theorem 2. *Let C_2' be a non-supersingular genus 2 hyperelliptic curve over $\mathbb{F}_p$ as above, with embedding degree $k>1$ and automorphisms ψ_2 and $\widehat{\psi}_2$ defined as above. Let $D_1=[u_1(x),v_1(x)]\in\mathcal{J}_{C_2'}(\mathbb{F}_p)$ be a reduced divisor of prime order n, where $n^2\nmid\#\mathcal{J}_{C_2'}(\mathbb{F}_p)$. Let $[\lambda_2]$ act as the multiplication map on the subgroup $\langle D_1\rangle$ defined as above such that $[\lambda_2]D_1=\psi_2(D_1)$. Suppose $m\in\mathbb{Z}$ is such that $mn=\lambda_2^4+1$. Then for $D_2\in\mathcal{J}_{C_2'}(\mathbb{F}_{p^k})$, we have*

$$\langle D_1,D_2\rangle_n^{\frac{m(p^k-1)}{n}} = \left[f_{\lambda_2,D_1}^{\lambda_2^3}(D_2)\cdot f_{\lambda_2,D_1}^{\lambda_2^2}\left(\widehat{\psi}_2(D_2)\right)\cdot f_{\lambda_2,D_1}^{\lambda_2}\left(\widehat{\psi}_2^2(D_2)\right)\cdot f_{\lambda_2,D_1}\left(\widehat{\psi}_2^3(D_2)\right)\cdot u_1(D_2)\right]^{\frac{p^k-1}{n}}.$$

Proof. The proof is similar to that of Theorem 1. Therefore, we omit it here. □

From Theorem 1 and Theorem 2, we note that the pairing computation on curve C_2' is more efficient than that on curve C_1'. Hence, the following discussions only focus on the curve C_2'.

4.2 A New Variant of Miller's Algorithm

In this subsection, we propose a new variant of Miller's algorithm for the family of genus 2 hyperelliptic curves C_2' over $\mathbb{F}_p$ with efficiently computable automorphisms ψ_2 and $\widehat{\psi}_2$. From Theorem 2, we obtain the following Algorithm 2 for computing the Tate pairing on such curves C_2', which is a variant of Miller's Algorithm (see Algorithm 1 in Section 2.2). For the curve C_1', we can also obtain a similar variant of Miller's algorithm as in Algorithm 2, based on Theorem 1.

5 Implementing the Tate Pairing with Efficient Automorphisms

In this section, we describe various techniques that enable the efficient implementation of the Tate pairing on a non-supersingular genus 2 curve of type C_2' over $\mathbb{F}_p$. Furthermore, we also analyze the computational cost of our new algorithm in detail and give timings for our implementation.

Algorithm 2. Computing the Tate Pairing with Efficient Automorphisms

IN: $\overline{D}_1 = [u_1, v_1] \in \mathcal{J}_{C_2'}(\mathbb{F}_p)[n], \overline{D}_2 \in \mathcal{J}_{C_2'}(\mathbb{F}_{p^k})$, represented by D_1 and D_2 with $\mathrm{supp}(D_1) \cap \mathrm{supp}(D_2) = \emptyset$, $\lambda_2 = (e_r, e_{r-1}, \ldots, e_0)_2$, where $e_i \in \{0, 1\}$ for $i = 0, \ldots, r-1$ and $e_r = 1$, and $mn = \lambda_2^4 + 1$.

OUT: $\langle D_1, D_2 \rangle_n^{m(p^k-1)/n}$

1. $T \leftarrow D_1, f_1 \leftarrow 1, f_2 \leftarrow 1, f_3 \leftarrow 1, f_4 \leftarrow 1, f_5 \leftarrow u_1(D_2)$

2. for i from $r - 1$ downto 0 do

3. ▷ Compute T' and $G_{T,T}(x, y)$ such that $T' = 2T - \mathrm{div}(G_{T,T})$

4. $\overline{T} \leftarrow [2]\overline{T}, f_1 \leftarrow f_1^2 \cdot G_{T,T}(D_2), f_2 \leftarrow f_2^2 \cdot G_{T,T}(\widehat{\psi}_2(D_2))$

5. $f_3 \leftarrow f_3^2 \cdot G_{T,T}(\widehat{\psi}_2^2(D_2)), f_4 \leftarrow f_4^2 \cdot G_{T,T}(\widehat{\psi}_2^3(D_2))$

6. if $e_i = 1$ then

7. ▷ Compute $T^{'}$ and $G_{T,D_1}(x, y)$ such that $T^{'} = T + D_1 - \mathrm{div}(G_{T,D_1})$

8. $\overline{T} \leftarrow \overline{T} \oplus \overline{D}_1, f_1 \leftarrow f_1 \cdot G_{T,D_1}(D_2), f_2 \leftarrow f_2 \cdot G_{T,D_1}(\widehat{\psi}_2(D_2))$

9. $f_3 \leftarrow f_3 \cdot G_{T,D_1}(\widehat{\psi}_2^2(D_2)), f_4 \leftarrow f_4 \cdot G_{T,D_1}(\widehat{\psi}_2^3(D_2))$

10. $f \leftarrow ((f_1^{\lambda_2} \cdot f_2)^{\lambda_2} \cdot f_3)^{\lambda_2} \cdot f_4 \cdot f_5$

11. $f \leftarrow f^{(p^k-1)/n}$

12. Return f

5.1 Curve Generation

While generating suitable parameters for supersingular genus 2 hyperelliptic curves over prime fields is easy, it seems to be more difficult to generate pairing-friendly non-supersingular genus 2 curves over $\mathbb{F}_p$ because of the complicated algebraic structure of hyperelliptic curves. Only a few results have appeared in the literature to address this issue [12,16,23,25] and there is ongoing research in this direction. In [12], Freeman proposed the first explicit construction of pairing-friendly genus 2 hyperelliptic curves over prime fields with ordinary Jacobians by modeling on the Cocks-Pinch method for the elliptic curve case [8]. In the appendix of [12], we find two examples which belong to the curve family C_1' considered in this paper. Unfortunately, the curve parameters in the two examples are too large to be optimal for efficient implementation. In a recent paper [25], Kawazoe and Takahashi presented two different approaches for explicitly constructing pairing-friendly genus 2 curves of the type $y^2 = x^5 + ax$ over $\mathbb{F}_p$. One is an analogue of the Cocks-Pinch method and the other is a cyclotomic method. Their findings are based on the closed formulae [15,19] for the order of the Jacobian of hyperelliptic curves of the above type. In this paper we will restrict to the case $p \equiv 1 \pmod 8$ and generate a suitable non-supersingular pairing-friendly genus 2 hyperelliptic curves C_2' with embedding degree 4 using the theorems in [25]. The reason that we only consider curves with embedding degree 4 in this section is to facilitate performance comparisons between supersingular and non-supersingular genus 2 curves. However, we would like to point out that non-supersingular curves with higher embedding degree are available from [25] and that our method is also applicable to such curves.

To find good curve parameters which are suitable for applying our new algorithm, we use the following searching strategies. From Theorem 2 we note that the subgroup order n should satisfy $mn = \lambda_2^4 + 1$ for an integer m. Assume that we require the (160/1024) bit security level. Then n is a prime around 160 bits and λ_2 is at least 40 bits. Furthermore, since the bit length of λ_2 determines the length of the loop in Algorithm 2, λ_2 should be taken as small as possible. Based on these observations, we first check all λ_2's of the form $\lambda_2 = 2^a, a \in \{41, 42, \ldots, 60\}$. We found two λ_2's, namely $\lambda_2 = 2^{58}$ and 2^{59}, for which $\lambda_2^4 + 1$ has a prime factor of 164 bits and 162 bits, respectively. However, using the above two primes as the subgroup order n and running the algorithms of [25], we cannot find any curve. Therefore, we consider the slightly more complicated choice of $\lambda_2 = 2^a + 2^b$, where $a, b \in \{41, 42, \ldots, 50\}$ and $a > b$. After a couple of trials, we found that choosing $\lambda_2 = 2^{43} + 2^{10}$ generates a non-supersingular pairing-friendly genus 2 hyperelliptic curve whose Jacobian has embedding degree 4 with respect to a 163-bit prime n. The curve is given by the equation

$$C_2^* : y^2 = x^5 + 9x$$

over $\mathbb{F}_p$, for a 329-bit prime p, where the hexadecimal representations of n and p are as follows:

$n =$ `00000006 a37991af 81ddfa3a ead6ec83 1ca0fc44 75d5add9 (163 bits)`

$p =$ `0000016b 953ca333 acf202b3 0476f30f ff085473 6d0a0be4`
`c542fa48 66e5afba 7bc6cd6d 21ca9fad eef796f1 (329 bits)`

In the following five subsections, we will detail various techniques required to efficiently implement the calculation of the Tate pairing on the curve C_2^*.

5.2 Finite Field Arithmetic

As the embedding degree of the curve C_2^* in our implementation is $k = 4$, we first discuss how to construct the finite field $\mathbb{F}_{p^4}$. Rather than construct $\mathbb{F}_{p^4}$ as a direct quartic extension of $\mathbb{F}_p$, the best way is to define the field $\mathbb{F}_{p^4}$ as a quadratic extension of $\mathbb{F}_{p^2}$, which is in turn a quadratic extension of $\mathbb{F}_p$. Since the p is congruent to 5 modulo 12 in our implementation, the field $\mathbb{F}_{p^2}$ can be constructed by the irreducible binomial $x^2 + 3$ and the field $\mathbb{F}_{p^4}$ can be constructed as a quadratic extension of $\mathbb{F}_{p^2}$ by the irreducible binomial $x^2 - \sqrt{-3}$. Letting $\beta = -3$, elements of the field $\mathbb{F}_{p^2}$ can be represented as $a + b\sqrt{\beta}$, where $a, b \in \mathbb{F}_p$, whereas elements of the field $\mathbb{F}_{p^4}$ can be represented as $c + d\sqrt[4]{\beta}$, where $c, d \in \mathbb{F}_{p^2}$. Under this tower construction, a multiplication of two elements and a squaring of one element in $\mathbb{F}_{p^4}$ cost $9M$ and $6M$ in $\mathbb{F}_p$, respectively [21].

5.3 Encapsulated Group Operations

In [11], Fan *et. al.* proposed a method to encapsulate the computation of the line function with the group operations for genus 2 hyperelliptic curves over prime fields, and derived new mix-addition and doubling formulae in projective and new (weighted projective) coordinates, respectively. Applying their explicit formulae to the curve C_2^* defined above, we can respectively calculate the divisor class addition and doubling with $36M + 5S$ and $32M + 6S$ in $\mathbb{F}_p$ in new coordinates. We also include their explicit formulae in the appendix with some modifications for the curve C_2^*.

5.4 Using Degenerate Divisors and Denominator Elimination

For a hyperelliptic curve of genus $g > 1$, using a degenerate divisor as the image divisor is more efficient than using a general divisor when evaluating line functions. Frey and Lange [13] discussed in detail when it is permissible to choose a degenerate divisor as the second argument of Miller's algorithm. They also note that, when the embedding degree k is even, one might choose the second pairing argument from a set $S = \{(x, y) \in C(\mathbb{F}_{q^k}) \mid x \in \mathbb{F}_{q^{k/2}}, y \in \mathbb{F}_{q^k} \backslash \mathbb{F}_{q^{k/2}}\}$. Note that the point in the set S is on the quadratic twist of $C/\mathbb{F}_{q^{k/2}}$. When considering C_2^* as a curve defined over $\mathbb{F}_{p^2}$, we can define a quadratic twist of C_2^* over $\mathbb{F}_{p^2}$, denoted by $C_{2,t}^*$, as follows

$$C_{2,t}^* : y^2 = x^5 + 9c^4 x,$$

where $c \in \mathbb{F}_{p^2}$ is a quadratic non-residue over $\mathbb{F}_{p^2}$. It is known that $C_{2,t}^*(\mathbb{F}_{p^4}) \cong C_2^*(\mathbb{F}_{p^4})$ and the isomorphism of $C_{2,t}^*(\mathbb{F}_{p^4})$ and $C_2^*(\mathbb{F}_{p^4})$ also induces an isomorphism ϕ of $\mathcal{J}_{C_{2,t}^*}(\mathbb{F}_{p^4})$ and $\mathcal{J}_{C_2^*}(\mathbb{F}_{p^4})$ [26]. As in [11] we first generate a degenerate divisor class $\overline{D}_t = [x - x_t, y_t] \in \mathcal{J}_{C_{2,t}^*}(\mathbb{F}_{p^2})$ on the twisted curve $C_{2,t}^*/\mathbb{F}_{p^2}$. Then the isomorphism ϕ will map $\overline{D}_t$ to a degenerate divisor class $\overline{D}_2 = \phi(\overline{D}_t) = [x - c^{-1}x_t, c^{-5/2}y_t] \in \mathcal{J}_{C_2^*}(\mathbb{F}_{p^4})$ on the curve C_2^* over $\mathbb{F}_{p^4}$. Note that the denominator elimination technique applies in this case since $x - c^{-1}x_t$ is defined over $\mathbb{F}_{p^2}$. Furthermore, we do not need to compute $f_5 = u_1(D_2) \in \mathbb{F}_{p^2}$ in Algorithm 2 either, for the same reason.

5.5 Evaluating Line Functions at a Degenerate Divisor

Here we consider the evaluation of line functions at a degenerate divisor $D_2 = [x - x_2, y_2] \in \mathcal{J}_{C_2^*}(\mathbb{F}_{p^4})$ generated by the method in Section 5.4, where $x_2 = c^{-1}x_t \in \mathbb{F}_{p^2}$ and $y_2 = c^{-5/2}y_t \in \mathbb{F}_{p^4} \backslash \mathbb{F}_{p^2}$. Moreover, we further assume that in this work $c = \sqrt{-3}$ is taken as a quadratic non-residue over $\mathbb{F}_{p^2}$. Therefore, y_2 has only two non-zero coefficients instead of four in a polynomial basis representation of $\mathbb{F}_{p^4}$. Furthermore, since the denominator elimination technique applies in this case, we only need to evaluate the numerators of the rational functions at D_2. From [11] we know that in new coordinates we can respectively work with the numerators $c_1(x, y) = (Z_{31}Z_{32})y - ((s_1 z_{11})x^3 + l_2 x^2 + l_1 x + l_0)$ for group doubling and $c_2(x, y) = (\tilde{r} z_{21})y - ((s_1' z_{21})x^3 + l_2 x^2 + l_1 x + l_0)$ for group addition, where $Z_{31}, Z_{32}, \tilde{r}, z_{11}, z_{21}, s_1, s_1', l_2, l_1$ and l_0 are from Table 4 and Table 5 in the appendix. Note that in Algorithm 2 we need to evaluate the function $c_i(x, y), i = 1$

or 2 at four image divisors $D_2 = [x - x_2, y_2]$, $\widehat{\psi}_2(D_2) = [x - \xi_8^{-2} x_2, \xi_8^{-1} y_2]$, $\widehat{\psi}_2^2(D_2) = [x - \xi_8^4 x_2, \xi_8^{-2} y_2] = [x + x_2, \xi_8^{-2} y_2]$ and $\widehat{\psi}_2^3(D_2) = [x - \xi_8^2 x_2, \xi_8^{-3} y_2]$ for each iteration of the loop. Hence we have the following relations

$$\begin{aligned}
c_i(D_2) &= (\tilde{r} z_{11}) y_2 - [((s_1' z_{11}) x_2^3 + l_1 x_2) + (l_2 x_2^2 + l_0)],\\
c_i(\widehat{\psi}_2(D_2)) &= ((\tilde{r} z_{11}) y_2) \xi_8^{-1} - [((s_1' z_{11}) x_2^3 - l_1 x_2) \xi_8^2 - (l_2 x_2^2 - l_0)],\\
c_i(\widehat{\psi}_2^2(D_2)) &= ((\tilde{r} z_{11}) y_2) \xi_8^{-2} + [((s_1' z_{11}) x_2^3 + l_1 x_2) - (l_2 x_2^2 + l_0)],\\
c_i(\widehat{\psi}_2^3(D_2)) &= ((\tilde{r} z_{11}) y_2) \xi_8^{-3} + [((s_1' z_{11}) x_2^3 - l_1 x_2) \xi_8^2 + (l_2 x_2^2 - l_0)].
\end{aligned}$$

We assume that x_2^2, x_2^3, ξ_8^{-1} and ξ_8^2 are precomputed with $7M + 2S$ over $\mathbb{F}_p$. Since x_2, x_2^2 and x_2^3 are in $\mathbb{F}_{p^2}$ and y_2 has only two non-zero coefficients in the polynomial basis representation of $\mathbb{F}_{p^4}$, $c_i(D_2)$ can be computed with $10M$ over $\mathbb{F}_p$. By reusing the intermediate computation results, we can compute $c_i(\widehat{\psi}_2(D_2))$, $c_i(\widehat{\psi}_2^2(D_2))$ and $c_i(\widehat{\psi}_2^3(D_2))$ with $4M$, $2M$ and $2M$ over $\mathbb{F}_p$, respectively. Therefore, the total cost of evaluating the function $c_i(x, y)$ at the degenerate divisor D_2 is $18M$ over $\mathbb{F}_p$ per iteration, with a precomputation of $7M + 2S$. For the case of evaluating the rational functions at a general divisor, the reader is referred to [11].

5.6 Final Exponentiation

For a genus 2 curve with an embedding degree of $k = 4$, the output of Miller's algorithm needs to be raised to the power of $(p^4 - 1)/n$. Calculating this exponentiation can follow two steps as shown in [21]. Letting $f \in \mathbb{F}_{p^4}$ be the output of Miller's algorithm, the first step is to compute f^{p^2-1} which can be obtained with a conjugation with respect to $\mathbb{F}_{p^2}$ and $1I + 1M$ in $\mathbb{F}_{p^4}$. The remaining exponentiation to $(p^2 + 1)/n$ is an expensive operation which can be efficiently computed with the Lucas laddering algorithm [35] for the curve C_2^* in question.

5.7 Performance Analysis and Comparison

In this section, we analyze the computational complexity of the Algorithm 2 for calculating the Tate pairing on non-supersingular genus 2 hyperelliptic curves C_2' and compare the performance of pairing computations on supersingular and non-supersingular genus 2 curves over prime fields with the same embedding degree of $k = 4$.

We first analyze the algebraic complexity of the pairing computation on curves C_2' with our new algorithm (see Section 4.2). Recall that n is the subgroup order and λ_2 is a root of the equation $\lambda^4 + 1 \equiv 0 \bmod n$. We assume that the embedding degree k is even and the line functions in Algorithm 2 are evaluated at a degenerate divisor D_2 instead of a general divisor for efficiency reasons. We also assume that λ_2 has a random Hamming weight, meaning that about $\left(\frac{1}{2}\log_2 \lambda_2\right)$ additions take place in Algorithm 2 on average. Then the algebraic cost for computing the Tate pairing is given by (without including the cost of the final exponentiation)

$$T_1 + (\log_2 \lambda_2)(T_D + T_G + 4T_{sk} + 8T_{mk}) + \left(\frac{1}{2}\log_2 \lambda_2\right)(T_A + T_G + 8T_{mk}) + T_{10},$$

where

1. T_2 – the cost of precomputing f_5 in Step 1 of Algorithm 2.
2. T_D – the cost of doubling a general divisor.
3. T_A – the cost of adding two general divisors.
4. T_G – the cost of evaluating rational function $G(x, y)$ at the image divisors $D_2, \widehat{\psi}_2(D_2), \widehat{\psi}_2^2(D_2)$ and $\widehat{\psi}_2^3(D_2)$.
5. T_{sk} – the cost of squaring in $\mathbb{F}_{p^k}$.
6. T_{mk} – the cost of multiplication in $\mathbb{F}_{p^k}$.
7. T_{10} – the cost of computing f in Step 10 of Algorithm 2.

When applying various optimization techniques detailed in previous subsections to the particular curve C_2^*, we can further reduce the above cost of computing the Tate pairing to

$$43 \cdot (T_D + T_G + 4T_{sk} + 4T_{mk}) + (T_A + T_G + 4T_{mk}) + T_{10},$$

where $T_D = 32M + 6S, T_A = 36M + 5S, T_G = 18M, T_{sk} = 6M, T_{mk} = 9M$ and $T_{10} = 828M$. Furthermore, we also need $7M + 2S$ for precomputations (see Section 5.5). Note that all multiplications and squarings here are over $\mathbb{F}_p$. Therefore, the total cost of computing the Tate pairing with our optimizations is given by $5655M + 265S$ in $\mathbb{F}_p$.

In [6,11,21], the authors examined the implementation of the Tate pairing on a family of supersingular genus 2 hyperelliptic curves C_1 (see Section 3.1) over prime fields with embedding degree 4 in affine and projective coordinates, respectively. We compare the performance of pairing computations on the supersingular curve C_1 and the non-supersingular curve C_2^* in the following Table 1. Note that both curves have the same embedding degree of $k = 4$.

From Table 1, we note that for the same security level the computation of the Tate pairing on the non-supersingular genus 2 curve C_2^* is algebraically about 55.8% faster than on the supersingular genus 2 curve C_1, under the assumption that field squarings have cost $S = 0.8M$. Therefore, our algorithm improves previous work for pairing computations on genus 2 hyperelliptic curves over prime fields by a considerable margin.

Table 1. Performance Comparison of Pairing Computation on Curves C_1 and C_2^*

Reference	Curve Equation	Coordinate Type	Cost
Choie and Lee [6]	$C_1 : y^2 = x^5 + a,$ $a \in \mathbb{F}_p^*, p \equiv 2, 3 \bmod 5$	Affine	$240I, 17688M, 2163S$
Ó hÉigeartaigh & Scott [21]		Affine	$162I, 10375M, 645S$
Fan, Gong and Jao [11]		Projective	$13129M, 967S$
		New	$12487M, 971S$
This work	$C_2^* : y^2 = x^5 + 9x,$ $p \equiv 1 \bmod 8$	New	$\mathbf{5655M, 265S}$

5.8 Experimental Results

For verifying our theoretical analysis in Section 5.7, we report implementation results of computing the Tate pairing on the supersingular genus 2 curve C_1 and non-supersingular genus 2 curve C_2^* in this section. Both curves are defined over $\mathbb{F}_p$ and have an embedding degree of $k = 4$. The code was written in C and *Microsoft Developer Studio 6* was used for compilation and debugging on a Core 2 Duo$^{\text{TM}}$@2.67 GHz processor. For the curve C_1 and the (160/1024) bit security level we use the curve parameters that are generated in [21], where the subgroup order $n = 2^{159} + 2^{17} + 1$ is a Solinas prime [37] and the characteristic p of the finite field $\mathbb{F}_p$ is a 256-bit prime. Recall that the curve C_2^* is defined over a prime field of size 329 bits (see Section 5.1). The operations in the above two prime fields are implemented with various efficient algorithms in [20]. Table 2 shows the timings of our finite field library and the corresponding IM-ratio. From Table 2 we note that the IM-ratios are sufficiently large for the two prime fields in our implementation that using new coordinates and encapsulated group operations [11] can provide better performance than using affine coordinates in this case.

Table 3 summarizes previous work and our experimental results for the implementation of the Tate pairing on the curve C_1 and C_2^* for the (160/1024) bit security level. All of the timings are given in milliseconds.

From Table 3, we note that in our implementation the pairing computation on the curve C_2^* is about 10% faster than that on the curve C_1, in contrast to the algebraic complexity analysis in Section 5.7. The reason is that the sizes of the fields over which both curves are defined are different. Observe that the curve C_2^* is defined over a larger prime field than C_1, which significantly decreases the speed of computing the final exponentiation of the Tate pairing when the curve C_2^* is used. This explains why our new algorithm only obtains a 10% performance improvement in the implementation. Unfortunately, under current

Table 2. Timings of Prime Field $\mathbb{F}_p$ Library

Curve	# of bits of p	Multiplication (M)	Squaring (S)	Inversion (I)	IM-ratio
C_1	256	$0.84\mu s$	$0.78\mu s$	$41.9\mu s$	53.7
C_2^*	329	$1.40\mu s$	$1.30\mu s$	$64.6\mu s$	46.1

Table 3. Experimental Results – (160/1024) Security Level

Reference	Curve	Coordinate Type	Subgroup Order	Running Time (ms)
Choie and Lee [6]	C_1	Affine	Random	515
Ó hÉigeartaigh and Scott [21]	C_1	Affine	$n = 2^{159} + 2^{17} + 1$	16
This work	C_1	New	$n = 2^{159} + 2^{17} + 1$	14.6
	C_2^*	New	$\lambda_2 = 2^{43} + 2^{10}$	13.1

techniques for generating pairing-friendly non-supersingular genus 2 hyperelliptic curves, we cannot find such a curve of the form $y^2 = x^5 + ax$ defined over a 256-bit prime field with an embedding degree of $k = 4$. Nevertheless, despite the unequal field size, our implementation on the curve C_2^* is still slightly faster, even though strictly speaking a direct comparison between fields of different size is complicated as many factors could affect the comparison one way or another.

6 Conclusion

In this paper, we have proposed new variants of Miller's algorithm for computing the Tate pairing on two families of non-supersingular genus 2 hyperelliptic curves over prime fields with efficiently computable automorphisms. We describe how to use the automorphisms to unroll the main loop of Miller's algorithm. As a case study, we combine our new algorithm with various optimization techniques in the literature to efficiently implement the Tate pairing on a non-supersingular genus 2 curve $y^2 = x^5 + 9x$ over $\mathbb{F}_p$ with an embedding degree of $k = 4$. We also analyze the performance for the new algorithm in detail. When compared with pairing computations on supersingular genus 2 curves at the same security level, our new algorithm can obtain 55.8% performance improvements algebraically. Furthermore, favorable experimental results have been obtained for the implementation of the Tate pairing on both a supersingular and a non-supersingular genus 2 curve with embedding degree 4.

References

1. Avanzi, R.M., Cohen, H., Doche, C., Frey, G., Lange, T., Nguyen, K., Vercauteren, F.: Handbook of Elliptic and Hyperelliptic Curve Cryptography. Chapman & Hall/CRC, Boca Raton (2006)
2. Barreto, P.L.S.M., Galbraith, S., ÓhÉigeartaigh, C., Scott, M.: Efficient Pairing Computation on Supersingular Abelian Varieties. Design, Codes and Cryptography 42, 239–271 (2007)
3. Barreto, P.L.S.M., Kim, H.Y., Lynn, B., Scott, M.: Efficient Algorithm for Pairing-Based Cryptosystems. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 354–368. Springer, Heidelberg (2002)
4. Barreto, P.L.S.M., Lynn, B., Scott, M.: On the Selection of Pairing-Friendly Groups. In: Matsui, M., Zuccherato, R.J. (eds.) SAC 2003. LNCS, vol. 3006, pp. 17–25. Springer, Heidelberg (2004)
5. Boneh, D., Franklin, M.: Identity-Based Encryption from the Weil Pairing. SIAM Journal of Computing 32(3), 586–615 (2003)
6. Choie, Y., Lee, E.: Implementation of Tate Pairing on Hyperelliptic Curve of Genus 2. In: Lim, J.-I., Lee, D.-H. (eds.) ICISC 2003. LNCS, vol. 2971, pp. 97–111. Springer, Heidelberg (2004)
7. Choie, Y., Jeong, E., Lee, E.: Supersingular Hyperelliptic Curves of Genus 2 over Finite Fields. Journal of Applied Mathematics and Computation 163(2), 565–576 (2005)

8. Cocks, C., Pinch, R.G.E.: Identity-based Cryptosystems Based on the Weil Pairing (unpublished manuscript, 2001)
9. Duursma, I., Gaudry, P., Morain, F.: Speeding up the Discrete Log Computation on Curves with Automorphisms. In: Lam, K.-Y., Okamoto, E., Xing, C. (eds.) ASIACRYPT 1999. LNCS, vol. 1716, pp. 103–121. Springer, Heidelberg (1999)
10. Duursma, I., Lee, H.S.: Tate Pairing Implementation for Hyperelliptic Curves $y^2 = x^p - x + d$. In: Laih, C.-S. (ed.) ASIACRYPT 2003. LNCS, vol. 2894, pp. 111–123. Springer, Heidelberg (2003)
11. Fan, X., Gong, G., Jao, D.: Efficient Pairing Computation on Genus 2 Curves in Projective Coordinates. Centre for Applied Cryptographic Research (CACR) Technical Reports, CACR 2008-03, http://www.cacr.math.uwaterloo.ca/techreports/2008/cacr2008-03.pdf
12. Freeman, D.: Constructing Pairing-Friendly Genus 2 Curves over Prime Fields with Ordinary Jacobians. In: Takagi, T., Okamoto, T., Okamoto, E., Okamoto, T. (eds.) Pairing 2007. LNCS, vol. 4575, pp. 152–176. Springer, Heidelberg (2007)
13. Frey, G., Lange, T.: Fast Bilinear Maps from The Tate-Lichtenbaum Pairing on Hyperelliptic Curves. In: Hess, F., Pauli, S., Pohst, M. (eds.) ANTS 2006. LNCS, vol. 4076, pp. 466–479. Springer, Heidelberg (2006)
14. Frey, G., Rück, H.-G.: A Remark Concerning m-Divisibility and the Discrete Logarithm Problem in the Divisor Class Group of Curves. Mathematics of Computation 62(206), 865–874 (1994)
15. Furukawa, E., Kawazoe, M., Takahashi, T.: Counting Points for Hyperelliptic Curves of Type $y^2 = x^5 + ax$ over Finite Prime Fields. In: Matsui, M., Zuccherato, R.J. (eds.) SAC 2003. LNCS, vol. 3006, pp. 26–41. Springer, Heidelberg (2004)
16. Galbraith, S.D., McKee, J.F., Valença, P.C.: Ordinary Abelian Varieties Having Small Embedding Degree. Finite Fields and Their Applications 13(4), 800–814 (2007)
17. Gaudry, P.: An Algorithm for Solving the Discrete Log Problem on Hyperelliptic Curves. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 19–34. Springer, Heidelberg (2000)
18. Granger, R., Hess, F., Oyono, R., Thériault, N., Vercauteren, F.: Ate Pairing on Hyperelliptic Curves. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 430–447. Springer, Heidelberg (2007)
19. Haneda, M., Kawazoe, M., Takahashi, T.: Suitable Curves for Genus-4 HEC over Prime Fields: Point Counting Formulae for Hyperelliptic Curves of Type $y^2 = x^{2k+1} + ax$. In: Caires, L., Italiano, G.F., Monteiro, L., Palamidessi, C., Yung, M. (eds.) ICALP 2005. LNCS, vol. 3580, pp. 539–550. Springer, Heidelberg (2005)
20. Hankerson, D., Menezes, A., Vanstone, S.: Guide to Elliptic Curve Cryptography. Springer, New York (2004)
21. ÓhÉigeartaigh, C., Scott, M.: Pairing Calculation on Supersingular Genus 2 Curves. In: Biham, E., Youssef, A.M. (eds.) SAC 2006. LNCS, vol. 4356, pp. 302–316. Springer, Heidelberg (2007)
22. Hess, F., Smart, N.P., Vercauteren, F.: The Eta Pairing Revisited. IEEE Transactions on Information Theory 52(10), 4595–4602 (2006)
23. Hitt, L.: Families of Genus 2 Curves with Small Embedding Degree, Cryptology ePrint Archive, Report 2007/001 (2007), http://eprint.iacr.org/2007/001
24. Joux, A.: A One-Round Protocol for Tripartite Diffie-Hellman. In: Bosma, W. (ed.) ANTS 2000. LNCS, vol. 1838, pp. 385–394. Springer, Heidelberg (2000)

25. Kawazoe, M., Takahashi, T.: Pairing-friendly Hyperelliptic Curves of Type $y^2 = x^5 + ax$, Cryptology ePrint Archive, Report 2008/026 (2008) http://eprint.iacr.org/2008/026
26. Kozaki, S., Matsuo, K., Shimbara, Y.: Skew-Frobenius Maps on Hyperelliptic Curves. In: The 2007 Symposium on Cryptography and Information Security - SCIS 2007, IEICE Japan, pp. 1D2–4 (January 2007)
27. Lee, E., Lee, H.-S., Lee, Y.: Eta Pairing Computation on General Divisors over Hyperelliptic Curves $y^2 = x^7 - x \pm 1$. In: Takagi, T., Okamoto, T., Okamoto, E., Okamoto, T. (eds.) Pairing 2007. LNCS, vol. 4575, pp. 349–366. Springer, Heidelberg (2007)
28. Menezes, A., Okamoto, T., Vanstone, S.A.: Reducing Elliptic Curve Logarithms to a Finite Field. IEEE Transactions on Information Theory 39(5), 1639–1646 (1993)
29. Miller, V.S.: Short Programs for Functions on Curves (unpublished manuscript, 1986), http://crypto.stanford.edu/miller/miller.pdf
30. Miller, V.S.: The Weil Pairing and Its Efficient Calculation. Journal of Cryptology 17(4), 235–261 (2004)
31. Park, Y.-H., Jeong, S., Lim, J.: Speeding Up Point Multiplication on Hyperelliptic Curves with Efficiently-Computable Endomorphisms. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 197–208. Springer, Heidelberg (2002)
32. Rubin, K., Silverberg, A.: Supersingular Abelian Varieties in Cryptography. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 336–353. Springer, Heidelberg (2002)
33. Sakai, R., Ohgishi, K., Kasahara, M.: Cryptosystems Based on Pairings. In: Proceedings of the 2000 Symposium on Cryptography and Information Security - SCIS 2002, Okinawa, Japan, pp. 26–28 (2000)
34. Scott, M.: Faster Pairings Using an Elliptic Curve with an Efficient Endomorphism. In: Maitra, S., Veni Madhavan, C.E., Venkatesan, R. (eds.) INDOCRYPT 2005. LNCS, vol. 3797, pp. 258–269. Springer, Heidelberg (2005)
35. Scott, M., Barreto, P.L.S.M.: Compressed Pairings. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 140–156. Springer, Heidelberg (2004)
36. Silverman, J.H.: The Arithmetic of Elliptic Curves. Graduate Texts in Mathematics 106. Springer, Heidelberg (1986)
37. Solinas, J.: Generalized Mersenne Primes, Centre for Applied Cryptographic Research (CACR) Technical Reports, CORR 99-39, http://www.cacr.math.uwaterloo.ca/techreprots/1999/corr99-39.pdf
38. Takashima, K.: Scaling Security of Elliptic Curves with Fast Pairing Using Efficient Endomorphism. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Science E90-A(1), 152–159 (2007)
39. Zhao, C., Zhang, F., Huang, J.: Speeding Up the Bilinear Pairings Computation on Curves with Automorphisms, Cryptology ePrint Archive, Report 2006/474 (2006), http://eprint.iacr.org/2006/474

Appendix: Explicit Formulae for Genus 2 Curves over $\mathbb{F}_p$

In this appendix, we give efficient explicit formulae for group operations on genus 2 curves over $\mathbb{F}_p$ in new coordinates in the context of pairing computations. Table 4 and Table 5 address the cases of new coordinates. Given two divisor classes $\overline{E}_1$ and $\overline{E}_2$, Table 4 computes the divisor class $\overline{E}_3 = [u_3(x), v_3(x)]$ and the rational function $l(x)$ such that $E_1 + E_2 = E_3 + \mathrm{div}\left(\frac{y-l(x)}{u_3(x)}\right)$ in the new coordinate system, where $l(x) = \frac{s_1'}{rz_{23}}x^3 + \frac{l_2}{rz_{24}}x^2 + \frac{l_1}{rz_{24}}x + \frac{l_0}{rz_{24}}$. For doubling a reduced divisor class E_1, Table 5 calculates the divisor class $\overline{E}_3 = [u_3(x), v_3(x)]$ and the rational function $l(x)$ such that $2E_1 = E_3 + \mathrm{div}\left(\frac{y-l(x)}{u_3(x)}\right)$ in projective coordinates, where $l(x) = \frac{s_1}{s_1'Z_{32}}x^3 + \frac{l_2}{Z_{31}Z_{32}}x^2 + \frac{l_1}{Z_{31}Z_{32}}x + \frac{l_0}{Z_{31}Z_{32}}$.

Table 4. Mixed-Addition Formula on a Genus 2 Curve over $\mathbb{F}_p$ (New Coordinates) [11]

Input	Genus 2 HEC $C : y^2 = x^5 + ax$ $\overline{E}_1 = [U_{11}, U_{10}, V_{11}, V_{10}, 1, 1, 1, 1]$ and $\overline{E}_2 = [U_{21}, U_{20}, V_{21}, V_{20}, Z_{21}, Z_{22}, z_{21}, z_{22}]$	
Output	$\overline{E}_3 = [U_{31}, U_{30}, V_{31}, V_{30}, Z_{31}, Z_{32}, z_{31}, z_{32}] = \overline{E}_1 \oplus \overline{E}_2$ $l(x)$ such that $E_1 + E_2 = E_3 + \mathrm{div}\left(\frac{y-l(x)}{u_3(x)}\right)$	
Step	Expression	Cost
1	**Compute resultant and precomputations:** $z_{23} = Z_{21}Z_{22}, z_{24} = z_{21}z_{23}, \tilde{U}_{11} = U_{11}z_{21}$ $\tilde{U}_{10} = U_{10}z_{21}, y_1 = \tilde{U}_{11} - U_{21}, y_2 = U_{20} - \tilde{U}_{10}$ $y_3 = U_{11}y_1, y_4 = y_2 + y_3, r = y_2y_4 + y_1^2U_{10}$	$7M, 1S$
2	**Compute almost inverse of u_2 mod u_1:** $inv_1 = y_1, inv_0 = y_4$	–
3	**Compute s':** $w_0 = V_{10}z_{24} - V_{20}, w_1 = V_{11}z_{24} - V_{21}, w_2 = inv_0w_0$ $w_3 = inv_1w_1, s_1' = y_1w_0 + y_2w_1, s_0' = w_2 - U_{10}w_3$	$7M$
4	**Precomputations:** $\tilde{r} = rz_{23}, R = \tilde{r}^2, Z_{31} = s_1'Z_{21}, Z_{32} = \tilde{r}Z_{21}$ $z_{31} = Z_{31}^2, z_{32} = Z_{32}^2, \tilde{s}_0' = s_0'z_{21}$	$4M, 3S$
5	**Compute l:** $l_2 = s_1'U_{21} + \tilde{s}_0', l_0 = s_0'U_{20} + rV_{20}$ $l_1 = (s_1' + s_0')(U_{21} + U_{20}) - s_1'U_{21} - s_0'U_{20} + rV_{21}$	$5M$
6	**Compute U_3:** $w_1 = \tilde{U}_{11} + U_{21}, U_{31} = s_1'(2\tilde{s}_0' - s_1'y_1) - z_{32}, l_1' = l_1s_1'$ $U_{30} = \tilde{s}_0'(s_0' - 2s_1'U_{11}) + s_1'^2(y_3 - \tilde{U}_{10} - U_{20}) + 2l_1' + Rw_1$	$7M, 1S$
7	**Compute V_3:** $w_1 = l_2s_1' - U_{31}, V_{30} = U_{30}w_1 - z_{31}(l_0s_1')$ $V_{31} = U_{31}w_1 + z_{31}(U_{30} - l_1')$	$6M$
Sum		$36M, 5S$

Table 5. Doubling Formula on a Genus 2 Curve over $\mathbb{F}_p$ (New Coordinates) [11]

Input	Genus 2 HEC $C : y^2 = x^5 + ax$ $\overline{E}_1 = [U_{11}, U_{10}, V_{11}, V_{10}, Z_{11}, Z_{12}, z_{11}, z_{12}]$	
Output	$\overline{E}_3 = [U_{31}, U_{30}, V_{31}, V_{30}, Z_{31}, Z_{32}, z_{31}, z_{32}] = [2]\overline{E}_1$ $l(x)$ such that $2E_1 = E_3 + \text{div}\left(\frac{y-l(x)}{u_3(x)}\right)$	
Step	Expression	Cost
1	**Compute resultant:** $w_0 = V_{11}^2, w_1 = U_{11}^2, w_2 = V_{10}z_{11}, w_3 = w_2 - U_{11}V_{11}$ $r = U_{10}w_0 + V_{10}w_3$	$4M, 2S$
2	**Compute almost inverse:** $inv_1' = -V_{11}, inv_0' = w_3$	–
3	**Compute k':** $\tilde{U}_{10} = U_{10}z_{11}, k_1' = z_{12}(2(w_1 - \tilde{U}_{10}) + w_1)$ $k_0' = (z_{12}U_{11})(4\tilde{U}_{10} - w_3) - w_0$	$4M$
4	**Compute s':** $w_0 = k_0' inv_0', w_1 = k_1' inv_1'$ $s_1' = w_2k_1' - V_{11}k_0', s_0' = w_0 - \tilde{U}_{10}w_1$	$5M$
5	**Precomputations:** $Z_{31} = s_1'z_{11}, z_{31} = Z_{31}^2, w_0 = rz_{11}, w_1 = w_0Z_{12}$ $Z_{32} = 2w_1Z_{11}, z_{32} = Z_{32}^2, w_2 = w_1^2, R = rZ_{31}$ $S_0 = s_0'^2, S = s_0'Z_{31}, s_0 = s_0's_1', s_1 = s_1'Z_{31}$	$8M, 4S$
6	**Compute l:** $l_2 = s_1U_{11} + s_0z_{11}, V_{10}' = RV_{10}$ $l_0 = s_0U_{10} + 2V_{10}', V_{11}' = RV_{11}$ $l_1 = (s_1 + s_0)(U_{11} + U_{10}) - s_1U_{11} - s_0U_{10} + 2V_{11}'$	$6M$
7	**Compute U_3:** $U_{30} = S_0 + 4(V_{11}' + 2w_2U_{11}), U_{31} = 2S - z_{32}$	$1M$
8	**Compute V_3:** $w_0 = l_2 - U_{31}, w_1 = w_0U_{30}, w_2 = w_0U_{31}$ $V_{31} = w_2 + z_{31}(U_{30} - l_1), V_{30} = w_1 - z_{31}l_0$	$4M$
Sum		$32M, 6S$