Biextensions in Pairing-based Cryptography

Jianming Lin[1], Damien Robert[3], Chang-An Zhao[1,2*], Yuhao Zheng[1]

[1]School of Mathematics, Sun Yat-sen University, Guangzhou, 510275, Guangdong, China.
[2]Guangdong Provincial Key Laboratory of Information Security Technology, Guangzhou, 510006, Guangdong, China.
[3]Inria Bordeaux, Institut de Mathématiques de Bordeaux, France.

*Corresponding author(s). E-mail(s): zhaochan3@mail.sysu.edu.cn;
Contributing authors: linjm28@mail2.sysu.edu.cn;
damien.robert@inria.fr; zhengyh57@mail2.sysu.edu.cn;

Abstract

Bilinear pairings constitute a cornerstone of public-key cryptography, where advancements in Tate pairings and their efficient variants have emerged as a critical research domain within cryptographic science. Currently, the computation of pairings can be effectively implemented through three distinct algorithmic approaches: Miller's algorithm, the elliptic net algorithm (as developed by Stange), and cubical-based algorithms (as proposed by Damien Robert). Biextensions are the geometric object underlying the arithmetic of pairings, and all three approaches can be seen as a different way to represent biextension elements. In this paper, we revisit the biextension geometric point of view for pairing computation and investigate in more detail the cubical representation for pairing-based cryptography. Utilizing the twisting isomorphism, we derive explicit formulas and algorithmic frameworks for the ate pairing and optimal ate pairing computations. Additionally, we present detailed formulas and introduce an optimized shared cubical ladder algorithm for super-optimal ate pairings. Through concrete computational analyses, we compare the performance of our cubical-based methods with the Miller's algorithm on various well-known families of pairing-friendly elliptic curves. Our results demonstrate that the cubical-based algorithm outperforms the Miller's algorithm by bits in certain specific situations, establishing its potential as an alternative for pairing computation.

Keywords: Pairing computation Miller's algorithm biextension cubical arithmetic super-optimal ate pairing

1 Introduction

In recent years, bilinear pairings have emerged as a crucial part of public-key cryptography, primarily owing to their applications in numerous protocols, such as identity-based encryption [5], short signatures [6], and zero-knowledge proofs [13, 12, 1]. A pairing is a non-degenerate bilinear map on an elliptic curve E of the following form

$$e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$$

where $\mathbb{G}_1$, $\mathbb{G}_2$ are two additive subgroups of E with prime order r, and $\mathbb{G}_T$ is a multiplicative subgroup of $\mathbb{F}_{p^k}^*$ also with order r, where k is the embedding degree of E.

In pairing-based cryptographic systems, the Weil and Tate pairings are commonly employed. In most cryptographic applications, The Tate pairing and its variants typically exhibit superior efficiency in practical implementations. Consequently, a substantial amount of research have focused on optimizing the Tate pairing and its variants. One of the research objectives is to shorten the length of the Miller loop. Duursma and Lee [11], along with Barreto *et al.* [3] have successfully shortened the length of the Miller iteration required for the Tate pairing on supersingular abelian varieties leveraging the η_T method. In 2006, Hess *et al.* [20] extended this idea to all ordinary curves through the application of the Frobenius endomorphism and proposed the **ate pairing**. Subsequently, several variants of the ate pairing [22, 25, 37] have been successively proposed, aiming to further minimize the length of the Miller iteration. Vercauteren introduced the notion of the **optimal (ate) pairing**, which can be computed using $\log_2 r/\varphi(k)$ basic Miller iterations, with φ representing the Euler totient function. When the underlying curve supports fast non-trivial automorphisms beyond Frobenius maps, the iteration length can be shortened to $\log_2 r/2\varphi(k)$ and the corresponding pairing is named **super-optimal (ate) pairing**. Recent advances in pairing computation have witnessed significant theoretical breakthroughs, with several works [29, 14, 10, 9, 23] establishing novel frameworks for super-optimal ate pairing implementations.

A significant direction of the research on accelerating the pairing computation is focused on enhancing the performance of the Miller iteration. All the efficient algorithms designed for computing the Tate pairing and its variants are based on Miller's algorithm [26]. Since then, a huge number of works [3, 4, 9, 10, 11, 16, 20, 22, 25, 31, 37] have enhanced the efficiency of this algorithm. Up to now, the Miller's algorithm still stands as the most effective approach for computing pairings.

Elliptic net algorithm (ENA) first proposed by Stange [33] is another method for computing pairings in polynomial time. In 2015, Chen *et al.* [8] optimized it by reducing the dimension of the blocks required in the algorithm, with an extra inversion at the DoubleAdd step. This improved variant is named IENA. Subsequently, Cai *et al.* [7] further strengthened the implementation of IENA, narrowing the performance gap between (I)ENA and Miller's algorithm. Nevertheless, the elliptic net algorithm is significantly less efficient compared to the Miller's algorithm.

Robert [30] presented a novel approach by leveraging cubical arithmetic to work in biextension for pairing computation, deriving highly efficient formulas on specific models of elliptic curves and Kummer lines. For generic pairings on Montgomery curves,

the cubical ladder algorithm obtained costs of only 15 field multiplications [30] per bit, which is faster than any pairing formula reported in the existing literature. This improvement benefits the implementation of numerous isogeny-based cryptographic schemes that necessitate generic pairing computations on the Montgomery model. However, there has been a lack of relevant research that deeply investigate the utilization of cubical arithmetic for pairing computations in elliptic curve cryptography (ECC) and make concrete cost analysis.

Biextensions, as introduced by Mumford in [28] and developped by Grothendieck in [18], are geometric objects that encode the arithmetic properties of pairings. All three approaches above to pairing computation can be seen as different ways to represent biextension elements. The use of biextensions as an algorithmic tool for pairings was, to the best of our knowledge, first investigated by Stange in her PhD thesis [32], where she explains in detail how elliptic nets formulas are a way to work with biextensions. These algorithmic aspects were further developed by Robert in [30], where the biextension interpretation of Miller's algorithm was given, and a new representation of biextension elements was given, called cubical arithmetic.

1.1 Contributions

In this paper, we reinvestigate the technique of biextension, applying it to pairing-based cryptography to derive more specific and efficient formulas for implementation. Besides, we make a detailed computational cost analysis and compare the performance of our proposed algorithms to that of Miller's algorithm. The key contributions of this paper are summarized as follows:

1. Biextension is the geometric object underlying pairings, in particular the biextension arithmetic is naturally bilinear. Many tools developed for scalar multiplication on elliptic curve have a natural generalization on biextension. In this paper we systematically develop this point of view, notably on the use of biextension twists and automorphisms to speed up the biextension arithmetic. We show how this geometric point of view allows to recover many pairing formulas in the literature (in the Miller representation), while providing greater conceptual clarity.
2. We then specialize the above geometric tools to the cubical representation. First, by employing twisting isomorphisms, we have derived more precise formulas and algorithms for the ate pairing and the optimal ate pairing computation through cubical arithmetic compared to those presented in [30]. Secondly, using efficiently-computable endomorphisms, we propose new efficient formulas for the super-optimal ate pairing. In addition, we present an optimized shared cubical ladder algorithm for the implementation.
3. We conduct a meticulous efficiency analysis for the algorithms in this paper. In particular, we investigate in details two different ways to perform biextension exponentiations in the cubical representation: via the cubical ladder, and via a double and add approach.

   Subsequently, we compared the efficiency of both methods with that of the Miller's algorithm on several well-known families of pairing-friendly curves. The results illustrate that the cubical arithmetic demonstrates better performance than

the Miller's algorithm in terms of a basic iteration by bits under certain specific circumstances (where the embedding degree k is an odd prime and the CM discriminant is 1), making it a possible competitive alternative of Miller's algorithm in pairing-based cryptography.

1.2 Organizations of this paper

The mathematical preliminaries and definitions are presented in Section 2. The Tate pairing and its variants used in pairing-based cryptography are recalled. Our theory and concrete formulas of pairings using biextension are stated in Section 3. Section 4 illustrates the concrete computational cost analysis and comparison. Finally, our conclusion are drawn in Section 5.

2 Preliminaries

In this section, we introduce the mathematical preliminaries and fundamental descriptions required in this paper. Let E denote an ordinary elliptic curve over a finite field $\mathbb{F}_p$, where $p > 5$ is a prime. Assume that E is a short Weierstrass curve. Then the rational points (x, y) with $x, y \in \mathbb{F}_p$ on E satisfy the following equation

$$E/\mathbb{F}_p : y^2 = x^3 + ax + b.$$

Define the point at infinity $\mathcal{O}_E$ to be the neutral element of $E(\mathbb{F}_p)$. The j-invariant of E is given by $j(E) = 1728 \cdot \frac{4a^3}{4a^3+27b^2}$. Denote by $\#E(\mathbb{F}_p)$ the cardinality of $E(\mathbb{F}_p)$. According to [36, Theorem 4.12], it holds that $\#E(\mathbb{F}_p) = p + 1 - t$, where t is the trace of the p-power Frobenius endomorphism $\pi : (x, y) \mapsto (x^p, y^p)$. Assume that E is a short Weierstrass curve in the remaining part of this paper.

Let r be a large prime divisor of $\#E(\mathbb{F}_p)$. The embedding degree k with respect to r is defined as the smallest positive integer such that $r \mid p^k - 1$. The three pairing subgroups $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ with order r are defined as follows

$$\begin{aligned}
\mathbb{G}_1 &= E[r] \cap \{P \in E \mid \pi(P) = P\} = E(\mathbb{F}_p)[r], \\
\mathbb{G}_2 &= E[r] \cap \{P \in E \mid \pi(P) = [p]P\}, \\
\mathbb{G}_T &= \mu_r \subseteq \mathbb{F}_{p^k},
\end{aligned}$$

where $E[r] = \{P \in E \mid [r]P = \mathcal{O}_E\}$ and μ_r are the r-torsion subgroup of E and the group of the r-th roots of unity, respectively. In the following, we introduce the definitions of the twist, endomorphism, bilinear pairing, together with biextension.

2.1 Twists and Endomorphisms of Elliptic Curves

Twisting isomorphisms and endomorphisms are two fundamental maps of elliptic curves that play a significant role in pairing-based cryptography by enhancing the implementation efficiency. In this subsection, we introduce the definitions and properties of these two morphisms.

Denote by $\mathrm{Aut}(E)$ the automorphism group of an elliptic curve E. Let $d = \#\mathrm{Aut}(E)$ represent the order of $\mathrm{Aut}(E)$. If d divides the embedding degree k, then E admits a degree-d twist E' defined over $\mathbb{F}_{p^e}$, where $e = k/d$ [20]. The map

$$\phi : E' \to E, \quad (x, y) \mapsto (\xi^2 x, \xi^3 y)$$

with $\xi \in \mathbb{F}_{p^k} \setminus \mathbb{F}_{p^e}$ is called the twisting isomorphism from E' to E, which implies that the two curves E and E' are isomorphic over $\mathbb{F}_{p^k}$. According to [20, Proposition 1], all twists corresponding to $\zeta \in \mathbb{F}_{p^e}^*/(\mathbb{F}_{p^e}^*)^d$ are given by

$$\begin{array}{lll} d = 2: & y^2 = x^3 + a/\zeta^2 x + b/\zeta^3, & \phi : E' \to E : (x, y) \mapsto (\zeta x, \zeta^{3/2} y), \\ d = 4: & y^2 = x^3 + a/\zeta x, & \phi : E' \to E : (x, y) \mapsto (\zeta^{1/2} x, \zeta^{3/4} y), \\ d = 3, 6: & y^2 = x^3 + b/\zeta, & \phi : E' \to E : (x, y) \mapsto (\zeta^{1/3} x, \zeta^{1/2} y). \end{array}$$

By employing the twisting isomorphism, the pairing subgroup $\mathbb{G}_2 \subseteq E(\mathbb{F}_{p^k})$ can be succinctly represented by the r-torsion subgroup of E'

$$\mathbb{G}_2 = E'[r] \cap \{P \in E \mid \pi(P) = [p]P\} \cong E'(\mathbb{F}_{p^{k/d}})[r].$$

We now consider the endomorphisms of E. Define D to be a positive square-free integer satisfying $4p - t^2 = Dy^2$, where $y \in \mathbb{Z}$. From [36, Theorem 10.6], the endomorphism ring of E over a finite field is isomorphic to an order in an imaginary quadratic field $\mathbb{Q}(\sqrt{-D})$. The maximal subring of $\mathbb{Q}(\sqrt{-D})$ is

$$\mathcal{O}_K = \begin{cases} \mathbb{Z}\left[\frac{1+\sqrt{-D}}{2}\right] & \text{if } D \equiv 3 \pmod 4, \\ \mathbb{Z}\left[\sqrt{-D}\right] & \text{if } D \equiv 1, 2 \pmod 4. \end{cases}$$

An order in $\mathbb{Q}(\sqrt{-D})$ is a ring R such that $\mathbb{Z} \subsetneq R \subseteq \mathcal{O}_K$ [36]. It can be expressed as

$$R = \mathbb{Z} + \mathbb{Z} f\delta$$

where $f > 0$ and $\delta = (1 + \sqrt{-D})/2$ or $\sqrt{-D}$. Let σ be an endomorphism of E over $\mathbb{F}_p$. Then σ can be conveniently represented as $\sigma = a + b\sqrt{-D}$, where $a, b \in \mathbb{Q}$ and $2a, 2b \in \mathbb{Z}$. If σ is a degree-n endomorphism, then the reduced norm of σ is $\mathrm{Nrd}(\sigma) = \sigma\bar{\sigma} = a^2 + b^2 D = n$. Besides, it satisfies the following characteristic equation

$$\sigma^2 - 2a\sigma + n = 0. \tag{1}$$

These endomorphisms allow for fast scalar multiplications via the GLV method [17]. Consequently, they are referred to as efficiently-computable endomorphisms, or simply GLV-endomorphisms. A curve equipped with such an endomorphism is denoted as a GLV-curve. If $n = 1$, then σ is naturally an automorphism.

In the following, we introduce two well-known GLV-curves over $\mathbb{F}_p$ with $D = 1$ and 3:

$$E_1 : y^2 = x^3 + b, \quad \text{where } p \equiv 1 \pmod 3,$$
$$E_2 : y^2 = x^3 + ax, \quad \text{where } p \equiv 1 \pmod 4.$$

There exists an automorphism $\sigma : (x, y) \mapsto (wx, y)$ on E_1, associated to $\frac{1+\sqrt{-3}}{2}$ in the endomorphism ring $\mathrm{End}_p(E_1)$, where w is a primitive cube root of unity in $\mathbb{F}_p^*$. According to Eq. (1), it satisfies $\sigma^2 + \sigma + 1 = 0$.

For E_2, the corresponding automorphism is $\sigma : (x, y) \mapsto (-x, iy)$, associated with $\pm\sqrt{-1}$ in $\mathrm{End}_p(E_2)$, where i is a primitive fourth root of unity in $\mathbb{F}_p^*$. This automorphism satisfies the characteristic equation $\sigma^2 + 1 = 0$.

2.2 Bilinear Pairings

In this subsection, we introduce some typical bilinear pairings used in ECC, including Tate pairings and their variants. With the notation as above, let E be an ordinary curve over $\mathbb{F}_p$. We first describe the definition of the Miller function.

For any point $P \in E$ and integer $n \in \mathbb{Z}$, let $f_{n,P}$ denote the normalized rational function associated with the divisor

$$\mathrm{div}(f_{n,P}) = n(P) - ([n]P) - (n-1)(\mathcal{O}_E).$$

In particular, for an r-torsion point $P \in E[r]$, the corresponding divisor is

$$\mathrm{div}(f_{r,P}) = r(P) - r(\mathcal{O}_E).$$

For all integers i, j, there exists a relationship between $f_{i,P}$, $f_{j,P}$, and $f_{i+j,P}$

$$\mathrm{div}(f_{i+j,P}) = \mathrm{div}\left(f_{i,P} \cdot f_{j,P} \cdot \frac{\ell_{[i]P,[j]P}}{v_{[i+j]P}}\right), \tag{2}$$

where $\ell_{[i]P,[j]P}$ represents the line passing through the points $[i]P$ and $[j]P$, and $v_{[i+j]P}$ represents the vertical line passing through $[i + j]P$ and $[-i - j]P$. A well-known efficient method for evaluating $f_{n,P}(Q)$ is the Miller's algorithm [26].

2.2.1 Tate pairing and its variants

Now we present the definitions of the Tate pairing and its variants. Let $P \in E(\mathbb{F}_{p^k})[r]$ and $Q \in E(\mathbb{F}_{p^k})$. The reduced Tate pairing is a non-degenerate bilinear map defined as follows

$$e_r : E(\mathbb{F}_{p^k})[r] \times E(\mathbb{F}_{p^k})/rE(\mathbb{F}_{p^k}) \to \mu_r, \quad (P, Q) \mapsto f_{r,P}(Q)^{\frac{p^k-1}{r}}.$$

By leveraging the efficiently-computable endomorphisms of E, one can reduce the length of the Miller loop. The ate pairing, as defined in [20], is an optimized variant of

the Tate pairing on $\mathbb{G}_2 \times \mathbb{G}_1$ and achieves a short Miller loop by employing the p-power Frobenius endomorphism π. Let P and Q be two points in $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. Denote by λ and m the two integers such that $\lambda \equiv p \mod r$ and $m = \frac{\lambda^k - 1}{r}$. The reduced ate pairing is presented as

$$a_\lambda : \mathbb{G}_2 \times \mathbb{G}_1 \to \mu_r, \quad (Q, P) \mapsto f_{\lambda,Q}(P)^{\frac{p^k-1}{r}},$$

which constitutes a non-degenerate bilinear map if $r \nmid m$. Several research [25, 37] have sought to further shorten the length of the Miller loop through multiplying or dividing the ate pairings.

Vercauteren [35] proposed an algorithm to construct optimal ate pairings, which can be computed in $\log_2(r)/\varphi(k)$ basic Miller iterations, where $\varphi(k)$ denotes the Euler function. Let $\lambda = mr$ such that $r \nmid m$. By Minkowski's theorem [27], there exists a short vector $V = (c_0, \cdots, c_{\varphi(k)-1})$, with $|c_i| \leq r^{1/\varphi(k)}$, satisfying $\lambda = \sum_{i=0}^{\varphi(k)-1} c_i p^i$. For points $P \in \mathbb{G}_1$ and $Q \in \mathbb{G}_2$, the optimal ate pairing [35] on E is defined as follows

$$\begin{aligned} &opt : \mathbb{G}_2 \times \mathbb{G}_1 \to \mu_r, \\ &(Q, P) \mapsto \left(\prod_{i=0}^{l} f_{c_i,Q}^{p^i}(P) \cdot \prod_{i=0}^{l-1} \frac{\ell_{[s_{i+1}]Q,[c_i p^i]Q}(P)}{v_{[s_i]Q}(P)} \right)^{(p^k-1)/r}, \end{aligned} \tag{3}$$

where $s_i = \sum_{j=i}^{l} c_j p^j$. This bilinear map is non-degenerate if

$$mkp^{k-1} \not\equiv \frac{p^k - 1}{r} \cdot \sum_{i=0}^{l} i c_i p^{i-1} \pmod r.$$

For specific families of pairing-friendly curves, the number of basic Miller iterations can be further reduced to $\log_2(r)/2\varphi(k)$. This type of pairing is named **super-optimal ate pairings** [34, 21, 29]. Such pairings [34, 21, 29, 14, 10, 9, 23] are constructed by compositing the power of Frobenius endomorphism and the GLV-endomorphism.

2.3 Biextensions

Biextensions were first introduced by Mumford in [28]. As mentioned in [30], biextensions provide a framework for studying pairings on abelian varieties. In this subsection, we focus primarily on biextensions associated with ordinary elliptic curves, and present the corresponding definitions, properties and the arithmetic.

Let $D = (\mathcal{O}_E)$ denote the polar divisor on an elliptic curve E, where $\mathcal{O}_E$ is the point at infinity. The biextension associated with this divisor, denoted by X_D, can be defined as follows.

Definition 1 ([30]). *Let D_P denote the divisor $(-P) - (\mathcal{O}_E)$. A biextension element is a tuple $(P, Q, g_{P,Q}) \in X_D$ where $P, Q \in E$, and $g_{P,Q}$ is a rational function with the divisor $D_{P+Q} + D_{\mathcal{O}_E} - D_P - D_Q$. Specifically,*

$$div(g_{P,Q}) = (-P - Q) + (\mathcal{O}_E) - (-P) - (-Q).$$

The function $g_{P,Q}$ is analogous to the line function (normalized at infinity) $\ell_{P,Q}$ that passes through points P and Q, as used in Miller iterations. For simplicity, we often omit P and Q and refer to an element of X_D simply as $g_{P,Q} \in \mathrm{X}_D$. The biextension X_D is equipped with two group laws, denoted by $\star_1$ and $\star_2$, which allow for the group addition law of elements. These operations are defined explicitly as follows

$$g_{P_1,Q} \star_1 g_{P_2,Q} = g_{P_1+P_2,Q} = g_{P_1,Q}(\cdot)g_{P_2,Q}(\cdot + P_1), \tag{4}$$

$$g_{P,Q_1} \star_2 g_{P,Q_2} = g_{P,Q_1+Q_2} = g_{P,Q_1}(\cdot)g_{P,Q_2}(\cdot)\frac{g_{Q_1,Q_2}(\cdot + P)}{g_{Q_1,Q_2}(\cdot)}. \tag{5}$$

These definitions ensure that the group laws respect the structure of the biextension and allow for a rich arithmetic framework. Since X_D is a symmetric biextension, we also have

$$g_{P_1,Q} \star_1 g_{P_2,Q} = g_{P_1+P_2,Q} = g_{P_1,Q}(\cdot)g_{P_2,Q}(\cdot)\frac{g_{P_1,P_2}(\cdot + Q)}{g_{P_1,P_2}(\cdot)}. \tag{6}$$

In accordance with the aforementioned additive group laws, we can formally define the inversion operation.

Definition 2 ([30]). *The inverse element $g_{P,Q}^{\star_1,-1}$ is formulated as*

$$g_{P,Q}^{\star_1,-1} = g_{-P,Q} = \frac{1}{g_{P,Q}} \cdot \frac{g_{-P,P}}{g_{-P,P}(\cdot + Q)}. \tag{7}$$

We note that the RHS does not depend on the choice of representative for $g_{-P,P}$. As per Definition 1 and Eq. (6), we deduce the following lemma, which elucidates the connection between $g_{P,Q}$ and the Miller function $f_{r,P}$. For further elaboration, refer to [30, Porism 3.10].

Lemma 1 ([30]). *Let $g_{P,Q} \in X_D$. Then, the Miller function $f_{r,-P}$ operating on the cycle $(\cdot) - (\cdot + Q)$ is given by $\frac{g_{[r]P,Q}(\cdot)}{g_{P,Q}^{r}(\cdot)}$.*

Proof. If $r = 0, 1$, the function $f_{r,-P}$ is constant, so its value on the cycle $(R)-(R+Q)$ is 1, which is also the value of the RHS. Now if we assume that the lemma is true for r_1, r_2, then by Eq. (2), we have

$$f_{r_1+r_2,-P} = \frac{f_{r_1,-P} \cdot f_{r_2,-P}}{g_{[r_1]P,[r_2]P}}$$

assuming all functions are normalized. Combining this equation with Eq. (6) deduces that the lemma is true for $r_1 + r_2$. □

From the point of view of Lemma 1, as explained by Grothendieck in [18], the biextension X_D is the intrinsic geometric object which encodes pairings (as monodromy in the biextension). The Miller functions $f_{r,P}$ are a way to compute the biextension arithmetic. But, like there are several ways to choose coordinates for an elliptic point

to do the arithmetic, we can also look at different representations of biextension elements for the biextension arithmetic. This is what we will do in Section 2.4 where we will use the cubical representation of biextension elements instead of the "Miller representation".

We remark that since $\mathcal{O}_E$ is a pole of $g_{P,Q}$, the value $g_{P,Q}(\mathcal{O}_E)$ is not well defined. Instead, we will always interpret it as an extended value with respect to the uniformizer x/y, i.e. as the value of the function $\frac{g_{P,Q}}{x/y}(\mathcal{O}_E)$. We say that the biextension element $g_{P,Q}$ is normalized if $g_{P,Q}(\mathcal{O}_E) = 1$. For instance, if $Q \neq -P$, we have that $g_{P,Q} = v_{-P-Q}/l_{-P,-Q}$ is the normalized biextension function above (P, Q). It follows that:

$$g_{P_1,Q} \star_1 g_{P_2,Q} = g_{P_1+P_2,Q} = g_{P_1,Q}(\cdot)g_{P_2,Q}(\cdot)\frac{v_{-P_1-P_2}}{l_{-P_1,-P_2}}(Q).$$

We also remark that if $Q = -P$, then $g_{P,-P} = 1/v_P$ is the normalized biextension function. We can also rewrite Lemma 1 as:

$$g_{[r]P,Q}(\mathcal{O}_E) = 1/f_{r,-P}(Q),$$

for $g_{P,Q}$ and $f_{r,-P}$ normalized.

We now look more closely at the biextension arithmetic, the Galois action on biextension elements, and biextension isomorphisms and twists. First, a biextension element $g_{P,Q}$ corresponds to the divisor $(-P-Q)+(\mathcal{O}_E)-(-P)-(-Q)$, so one can pick $g_{P,Q} = \frac{v_{P+Q}}{l_{-P,-Q}}$; this is the unique biextension element normalized at $\mathcal{O}_E$ with respect to the uniformizer x/y. In the special case where $Q = -P$, we instead take $g_{P,-P} = 1/v_P$.

Therefore, Eq. (7) can be rewritten as

$$\begin{aligned} g_{-P,Q}(\cdot) \coloneqq g_{P,Q}(\cdot)^{\star_1,-1} &= \frac{1}{g_{P,Q}(\cdot)} \cdot \frac{g_{-P,P}(\cdot)}{g_{-P,P}(\cdot+Q)} \\ &= \frac{1}{g_{P,Q}(\cdot)} \cdot \frac{\frac{1}{v_P(\cdot)}}{\frac{1}{v_P(Q+\cdot)}} \\ &= \frac{1}{g_{P,Q}(\cdot)} \cdot \frac{v_P(\cdot+Q)}{v_P(\cdot)}. \end{aligned}$$

Similarly, we can obtain

$$g_{P,-Q}(\cdot) = \frac{1}{g_{P,Q}}(\cdot) \cdot \frac{v_Q(\cdot+P)}{v_Q(\cdot)}.$$

It follows that

$$g_{-P,-Q}(\cdot) = g_{P,Q}(\cdot)\frac{v_P((\cdot+Q)-(\cdot))}{v_Q(\cdot+P)-(\cdot))}.$$

Plugging this formula into Lemma 1, we get

$$f_{r,P}((\cdot+Q)-(\cdot)) = \frac{g_{[r]P,Q}(\cdot)}{g^r_{P,Q}(\cdot)} \cdot \frac{v_P^r}{v_{[r]P}}((\cdot+Q)-(\cdot)). \tag{8}$$

In case of a final exponentiation, when P, Q, R all live in $\mathbb{F}_{p^k}$, this can be further simplified to

$$f_{r,P}((R+Q)-(R))^{(p^k-1)/r} = \left(\frac{g_{[r]P,Q}(R)}{v_{[r]P}((R+Q)-(R))}\right)^{(p^k-1)/r}.$$

As explained in Section 2.1 we will use twisting isomorphisms defined over some extension of $\mathbb{F}_p$ to speed up pairing computations. Biextension behave well with respect to isomorphisms.

Proposition 1. *Let $\phi : E_1 \to E_2$ be an isomorphism between two elliptic curves. Then*

$$\widetilde{\phi}: \; g_{P,Q} \mapsto g_{\phi^{-1}(P),\phi^{-1}(Q)} = \phi^{-1} \cdot g_{P,Q} = \phi^* g_{P,Q} = g_{P,Q} \circ \phi$$

is an isomorphism from the biextension on E_2 associated to $(\mathcal{O}_{E_2})$ to the biextension on E_1 associated to $(\mathcal{O}_{E_1})$.

Proof. This follows from the functoriality of biextensions. It can also be directly seen as follows: let $P' = \phi^{-1}(P), Q' = \phi^{-1}(Q)$, then $\phi^* g_{P,Q}$ has for divisor $(-P'-Q')+(\mathcal{O}_{E_1})-(P')-(Q')$ so is a biextension element above (P', Q'). Furthermore it is immediate from their definition that ϕ^* is compatible with the biextension laws $\star_1$ and $\star_2$. □

By using Proposition 1, one can use the isomorphism ϕ to do a biextension exponentiation in E_1 rather than E_2: go from $g_{P,Q}$ to $g_{P',Q'}$ using ϕ^*, do the biextension exponentiation in E_1, and go back to E_2 using $\phi^{-1,*}$.

We remark that if ϕ is not defined over the base field but over an extension, so that E_1 is a twist of E_2, then Proposition 1 realizes the biextension associated to E_1 as a twist of the biextension associated to E_2. This clarifies the usage of twisting isomorphisms in pairing based cryptography. Similarly to the case of elliptic curve arithmetic, where can be convenient to move to a twist; in the case of biextension arithmetic it can be convenient to move to a biextension twist. In both case the field of definition of the points (resp. biextension elements) may change. We will also use the Galois action:

Definition 3. *Let σ be an element of the Galois group of the base field k. Given a biextension element $g_{P,Q}$ we define a biextension element*

$$\sigma \cdot g_{P,Q} = g_{\sigma(P),\sigma(Q)} = \sigma \circ g_{P,Q} \circ \sigma^{-1}.$$

We will see in Section 3 how the Tate pairings only use the biextension arithmetic, while the ate/optimal ate pairings leverage the Galois action by the Frobenius element π_p to speed up the biextension arithmetic, and the super-optimal ate pairings also employ the action of automorphisms.

Indeed, by the arithmetic compatibility of biextensions, the biextension arithmetic is bilinear like:

$$(g_{P_1,Q_1} \star_1 g_{P_2,Q_1}) \star_2 (g_{P_1,Q_2} \star_1 g_{P_2,Q_2}) = (g_{P_1,Q_1} \star_2 g_{P_1,Q_2}) \star_1 (g_{P_2,Q_1} \star_2 g_{P_2,Q_2}).$$

In particular, biextension exponentiation is "bilinear" on the left and on the right with respect to $\star_1, \star_2$:

$$\begin{aligned}(g_{P_1,Q} \star_1 g_{P_2,Q})^{\star_1,n} &= g_{P_1,Q}^{\star_1,n} \star_1 g_{P_2,Q}^{\star_1,n},\\ (g_{P,Q_1} \star_2 g_{P,Q_2})^{\star_1,n} &= g_{P,Q_1}^{\star_1,n} \star_2 g_{P,Q_2}^{\star_1,n}.\end{aligned}$$

Besides, the Galois action σ and the automorphism ϕ^* also commute with $\star_1, \star_2$:

$$\begin{aligned}\sigma(g_{P_1,Q} \star_1 g_{P_2,Q}) &= (\sigma g_{P_1,Q}) \star_1 (\sigma g_{P_2,Q}),\\ \sigma(g_{P,Q_1} \star_2 g_{P,Q_2}) &= (\sigma g_{P,Q_1}) \star_2 (\sigma g_{P,Q_2}),\\ \phi^*(g_{P_1,Q} \star_1 g_{P_2,Q}) &= (\phi^* g_{P_1,Q}) \star_1 (\phi^* g_{P_2,Q}),\\ \phi^*(g_{P,Q_1} \star_2 g_{P,Q_2}) &= (\phi^* g_{P,Q_1}) \star_2 (\phi^* g_{P,Q_2}).\end{aligned}$$

So starting with a biextension function $g_{P,Q}$ and combining the three operations, we naturally obtain construction of biextension functions (above some points (P_2, Q_2)), construction which is bilinear in P, Q (for the $\star_1$ and $\star_2$ operations respectively). Now if $P_2 = \mathcal{O}_{E_2}$ or $Q_2 = \mathcal{O}_{E_2}$, the associated biextension function is constant, and the $\star$ operation reduces to standard multiplication in the base field. If the resulting constant function does not depend on the initial choice of $g_{P,Q}$ (i.e. depends only on P, Q), we have thus constructed a pairing. In fact, we only need that the result does not depend on the choice of $\mathbb{F}_{p^k}$-rational choice $g_{P,Q}$. The reason is that even though in practice we use normalized representatives, if $g_{P_1,Q}, g_{P_2,Q}$ are normalized, then $g_{P_1,Q} \star_1 g_{P_2,Q}$ will not be normalized in general, although it will be $\mathbb{F}_{p^k}$-rational if $P, Q \in E(\mathbb{F}_{p^k})$. Then by invariance we can replace $g_{P_1,Q} \star_1 g_{P_2,Q}$ by the normalized biextension function $g_{P_1+P_2,Q}$ and get the same result. In other words, biextension constructions are naturally bilinear, and the invariance under the choice of $\mathbb{F}_{p^k}$-rational representative ensure that the result is bilinear in P, Q. Reinterpreted through the lens of biextensions, we will see that all pairings construction in the litterature are of this form.

2.4 Cubical arithmetic for biextensions

In this section, we explore how to perform the biextension arithmetic, in particular biextension exponentiation, using the cubical arithmetic.

In particular, for the biextension function $g_{P,Q}$, we will look at the cubical representation as outlined in [30, Section 4.5]. An element $g_{P,Q} \in \mathrm{X}_D$ can be represented as

$$(P, Q, g_{P,Q}) = [\widetilde{P}, \widetilde{Q}; \widetilde{\mathcal{O}}_E, \widetilde{P+Q}],$$

where the first and last two components denote the poles and zeros of $g_{P,Q}$, respectively.

Here, $\widetilde{P}$ is a cubical point (of level 1) [30, Remark 4.32] represented by the cubical coordinate $Z_1(\widetilde{P})$. The biextension function $g_{P,Q}(\cdot)$ is then represented as a quotient of cubical functions

$$g_{P,Q}(R) = \frac{Z_1(\widetilde{R+P+Q})Z_1(\widetilde{R})}{Z_1(\widetilde{R+P})Z_1(\widetilde{R+Q})}, \tag{9}$$

where Z_1 is a choice of the sections of the divisor $D = (\mathcal{O}_E)$.

Here, the point $Z_1(\widetilde{R+P+Q})$ is evaluated via the cubical arithmetic, using the cube $\mathcal{O}_E, P, Q, R, Q+R, P+R, P+Q, P+Q+R$:

$$\frac{Z_1(\widetilde{P+Q+R})Z_1(\widetilde{P})Z_1(\widetilde{Q})Z_1(\widetilde{R})}{Z_1(\widetilde{\mathcal{O}_E})Z_1(\widetilde{Q+R})Z_1(\widetilde{P+R})Z_1(\widetilde{P+Q})} = g_{P,Q}(R)/g_{P,Q}(\mathcal{O}_E). \tag{10}$$

We remark that the RHS does not depend on the choice of biextension function $g_{P,Q}$ above (P,Q). Since $Z_1(\mathcal{O}_E) = 0$, the value $Z_1(\widetilde{\mathcal{O}}_E)$ should be understood as an extended value $(Z_1/(x/y))(\widetilde{\mathcal{O}}_E)$ with respect to the uniformizer x/y. We will always use the cubical point $\widetilde{\mathcal{O}}_E$ normalized to have $(Z_1/(x/y))(\widetilde{\mathcal{O}}_E) = 1$.

We can also use this cubical arithmetic to compute the biextension arithmetic:

Proposition 2. *Let $g_{P_1,Q}$ be represented by $[\widetilde{P_1}, \widetilde{Q}; \widetilde{\mathcal{O}}_E, \widetilde{P_1+Q}]$, and $g_{P_2,Q}$ be represented by $[\widetilde{P_2}, \widetilde{Q}; \widetilde{\mathcal{O}}_E, \widetilde{P_2+Q}]$. Then $g_{P_1+P_2,Q} = g_{P_1,Q} \star_1 g_{P_2,Q}$ is represented by $[\widetilde{P_1+P_2}, \widetilde{Q}; \widetilde{\mathcal{O}}_E, P_1 \widetilde{+ P_2+Q}]$, where $\widetilde{P_1+P_2}$ is an arbitrary cubical point above P_1+P_2 and $P_1 \widetilde{+ P_2+Q}$ is computed using the cubical law from Eq. (10).*

Proof. Let $g'_{P_1+P_2,Q}$ be the biextension function associated to

$$[\widetilde{P_1+P_2}, \widetilde{Q}; \widetilde{\mathcal{O}}_E, P_1 \widetilde{+ P_2+Q}]$$

by Eq. (9). It has the same divisor as $g_{P_1+P_2,Q}$, so it suffices to check the two functions agree on $\mathcal{O}_E$. Comparing Eq. (6) and Eq. (10), this is immediate. □

Corollary 1. *If $g_{P,Q}$ is represented by $[\widetilde{P}, \widetilde{Q}; \widetilde{\mathcal{O}}_E, \widetilde{P+Q}]$, then $g_{[r]P,Q} = g_{P,Q}^{\star_1;r}$ is represented by $[\widetilde{rP}, \widetilde{Q}; \widetilde{\mathcal{O}}_E, \widetilde{rP+Q}]$, and in particular:*

$$g_{[r]P,Q}(R) = \frac{Z_1(\widetilde{R}+[r]\widetilde{P}+\widetilde{Q})Z_1(\widetilde{R})}{Z_1(\widetilde{R}+[r]\widetilde{P})Z_1(\widetilde{R}+\widetilde{Q})}.$$

Combining with Lemma 1, this gives:

Corollary 2.

$$\begin{aligned}\frac{f_{r,-P}(R)}{f_{r,-P}(R+Q)} &= \frac{g_{[r]P,Q}(R)}{g_{P,Q}(R)^r} \\ &= \frac{Z_1(\widetilde{R}+[r]\widetilde{P}+\widetilde{Q})Z_1(\widetilde{R})}{Z_1(\widetilde{R}+[r]\widetilde{P})Z_1(\widetilde{R}+\widetilde{Q})} \cdot \left(\frac{Z_1(\widetilde{R}+\widetilde{P})Z_1(\widetilde{R}+\widetilde{Q})}{Z_1(\widetilde{R}+\widetilde{P}+\widetilde{Q})Z_1(\widetilde{R})}\right)^r.\end{aligned}$$

We will call a cubical point $\widetilde{P}$ normalized when $Z_1(\widetilde{P}) = 1$. If $\widetilde{P}, \widetilde{Q}, \widetilde{\mathcal{O}}_E, \widetilde{P+Q}$ are all normalized, then so is the associated biextension function $g_{P,Q}$. In particular, using these normalized points, if $f_{r,-P}$ is also normalized, we have

$$\frac{1}{f_{r,-P}(Q)} = g_{[r]P,Q}(\mathcal{O}_E) = \frac{Z_1([r]\widetilde{P} + \widetilde{Q})}{Z_1([r]\widetilde{P})}.$$

Remark 1 (Level 2 cubical arithmetic)**.** *For our algorithms, it will be convenient to switch to cubical points of level* 2 *[30, Remark 4.32]. We let* $Z = Z_1^2$*, this is a section of* $2D = 2(\mathcal{O}_E)$*, and* X *another section such that* $x = X/Z$*. A level 2 cubical point* $\widetilde{P}$ *is then determined by* $\widetilde{P} = (X(\widetilde{P}), Z(\widetilde{P})))$*. Working with level* 2 *cubical points means that we encode level* 2 *biextension functions, that is elements of the biextension* X_{2D} *associated to* $2D$*. The biextension arithmetic will thus compute the square of the usual pairings. When there is an ambiguity, we will use the notation* $g_{D,P,Q}$ *to specify the divisor we are working with on the biextension. For instance, given a biextension element* $g_{D,P,Q}$ *for* X_D*, then we have a biextension element* $g_{2D,P,Q} = g_{D,P,Q}^2$ *for* X_{2D}*.*

If $R = \mathcal{O}_E$*, a direct evaluation of* $g_{2D,P,Q}$ *at* R *is inadvisable since the point at infinity constitutes a zero of* $g_{P,Q}$*. According to [30, Remark 2.8], an extended value is required for this special case. Since we are in level* 2*, the extended value is given by* $(Z/(x/y)^2)(\widetilde{\mathcal{O}}_E) = (X/(x^3/y^2))(\widetilde{\mathcal{O}}_E) = X(\widetilde{\mathcal{O}}_E)$*, using that* $x = X/Z$ *and that* $y^2 = x^3 + ax + b$*. Recall that we define our neutral cubical point* $\widetilde{\mathcal{O}}_E$ *to be normalized. By the above computation we thus have* $X_{\mathcal{O}_E} = 1$*, and:*

$$g_{2D,P,Q}(\mathcal{O}_E) = \frac{Z_{P+Q} \cdot X_{\mathcal{O}_E}}{Z_P \cdot Z_Q}. \tag{11}$$

On this basis, if P *is an* r*-torsion point, the evaluation of* Z *at* $[r]P$ *also yields* $X_{[r]P}$*. Thus, it follows that*

$$g_{2D,[r]P,Q}(\mathcal{O}_E) = \frac{Z_{[r]P+Q} \cdot X_{\mathcal{O}_E}}{X_{[r]P} \cdot Z_Q}.$$

Given that $\frac{X_{[r]P+Q}}{X_Q} = \frac{Z_{[r]P+Q}}{Z_Q}$*, the function* $g_{2D,[r]P,Q}$ *can be expressed alternatively as*

$$g_{2D,[r]P,Q} = \frac{X_{[r]P+Q} \cdot X_{\mathcal{O}_E}}{X_{[r]P} \cdot X_Q}.$$

The coordinate values $Z_{[r]P+Q}$ and $Z_{[r]P}$ can be efficiently computed using the cubical ladder algorithm described in [30, Algorithm 4.2]. We now detail this. Suppose that we have a cubical representation of the biextension elements

$$g_{P_1,Q} = [\widetilde{P_1}, \widetilde{Q}, \widetilde{\mathcal{O}}_E, \widetilde{P_1 + Q}],\ g_{P_2,Q} = [\widetilde{P_2}, \widetilde{Q}, \widetilde{\mathcal{O}}_E, \widetilde{P_2 + Q}],$$
$$g_{P_1-P_2,Q} = [\widetilde{P_1 - P_2}, \widetilde{Q}, \widetilde{\mathcal{O}}_E, \widetilde{P_1 - P_2} + Q],$$

then we can compute $g_{P_1+P_2,Q} = [\widetilde{P_1+P_2}, \widetilde{Q}, \widetilde{\mathcal{O}}_E, P_1 \widetilde{+ P_2} + Q]$, via

$$\widetilde{P_1+P_2} = \texttt{cDIFF}(\widetilde{P_1}, \widetilde{P_2}, \widetilde{P_1-P_2}),$$
$$P_1 \widetilde{+ P_2} + Q = \texttt{cDIFF}(\widetilde{P_1+Q}, \widetilde{P_2}, P_1 \widetilde{- P_2} + Q),$$

where cDIFF denotes a cubical differential addition. In particular, we refer to Appendix A for explicit algorithms for cubical points of level 2 on curves with j-invariants 0 and 1728.

We remark that we only really need $\widetilde{P_2}$ from the cubical representation of $g_{P_2,Q}$ to perform the necessary operations. Furthermore, since the same cubical point $\widetilde{P_2}$ is used to compute $\widetilde{P_1+P_2}$ and $P_1 \widetilde{+ P_2} + Q$, we only require P_2: the resulting biextension element $g_{P_1+P_2,Q}$ does not depend on the choice of $\widetilde{P_2}$ above P_2.

This means that there are two strategies to compute a biextension exponentiation $g_{P,Q} \mapsto g_{[r]P,Q}$. Either we use a cubical ladder, computing $[n]\widetilde{P}, [n+1]\widetilde{P}, [n]\widetilde{P}+\widetilde{Q}$ with one cubical doubling and two cubical differential additions at each step, or a double-and-add ladder, keeping only $[n]\widetilde{P}, [n]\widetilde{P}+\widetilde{Q}$ at each step. When the current bit is 0, we execute a biextension doubling by computing $[2n]\widetilde{P}, [2n]\widetilde{P}+\widetilde{Q}$, which costs one cubical doubling and one cubical differential addition. When the current bit is 1, we first recover $[n+1]P = \texttt{cADD}([n]P, P, [n]P+Q, P-Q)$ using a compatible addition, and then we compute $[2n+1]\widetilde{P}, [2n+1]\widetilde{P}+\widetilde{Q}$ via two cubical differential additions. It is straightforward to extend the double-and-add method to incorporate windows and NAF. The compatible addition was introduced in [24], and we refer to Appendix A for explicit algorithms on curves with j-invariants 0 and 1728. We note also that once we have recovered $[n+1]P$ via the compatible addition, we can switch to the ladder approach, and conversely we can forget about $[n+1]\widetilde{P}$ in the ladder approach and switch to the double-and-add approach. This allows to switch dynamically between the two approaches, depending on whether the upcoming bits are successive 0s or not.

Remark 2. *In the double-and-add approach, it will often happen that we will need to use a compatible addition $P_1+P_2 = \texttt{cADD}(P_1, P_2, P_1+Q, P_2-Q)$ where P_1, P_2 lie in a smaller field k and Q lies in a bigger field k', hence P_1+Q, P_2+Q lie in k'. Then the compatible addition formulas will give $X_{P_1+P_2}, Z_{P_1+P_2}$ in the big field k'. One could go back to the small field k by computing $x(P_1+P_2) = X_{P_1+P_2}/Z_{P_1+P_2} \in k$, but this would require an expensive inversion.*

Instead, our strategy is to take any k-linear form $\psi : k' \to k$, and apply it to $(X_{P_1+P_2} = x(P_1+P_2)Z_{P_1+P_2}, Z_{P_1+P_2})$ to obtain $(x(P_1+P_2)\psi(Z_{P_1+P_2}), \psi(Z_{P_1+P_2}))$. This gives a projective representation in k of the coordinates of P_1+P_2 as long as $\psi(Z_{P_1+P_2}) \neq 0$.

Similar to biextensions, the cubical arithmetic behaves well with respect to isomorphisms.

Proposition 3. *Let $\phi : E_1 \to E_2$ be an isomorphism between two elliptic curves. Let $\widetilde{\phi}$ be the unique lift of ϕ to cubical points that sends $\widetilde{\mathcal{O}}_{E_1}$ to $\widetilde{\mathcal{O}}_{E_2}$. Then $\widetilde{\phi}$ is compatible with the cubical arithmetic.*

Proof. This follows from the unicity of the cubical torsor structure associated to a divisor on an elliptic curve. This can also be checked directly: let $Z_2 = Z_1 \circ \widetilde{\phi}$, where $\widetilde{\phi}$ is for now an arbitrary lift of ϕ. Then given a cube $\mathcal{O}_{E_2}, P, Q, R, Q+R, P+R, P+Q, P+Q+R$ on E_2, we have:

$$\frac{Z_2(\widetilde{P+Q+R})Z_2(\widetilde{P})Z_2(\widetilde{Q})Z_2(\widetilde{R})}{Z_2(\widetilde{\mathcal{O}_{E_2}})Z_2(\widetilde{Q+R})Z_2(\widetilde{P+R})Z_2(\widetilde{P+Q})} = g_{P,Q}(R)/g_{P,Q}(\mathcal{O}_{E_2}).$$

If we let $\widetilde{P'} = \widetilde{\phi}^{-1}(\widetilde{P}), \ldots$ and $g_{P',Q'} = \phi^* g_{P,Q}$ we find that we also have a cube on E_1:

$$\frac{Z_1(\widetilde{P'+Q'+R'})Z_1(\widetilde{P'})Z_1(\widetilde{Q'})Z_1(\widetilde{R'})}{Z_1(\widetilde{\phi}^{-1}(\widetilde{\mathcal{O}_{E_2}}))Z_1(\widetilde{Q'+R'})Z_1(\widetilde{P'+R'})Z_1(\widetilde{P'+Q'})} = g_{P',Q'}(R')/g_{P',Q'}(\mathcal{O}_{E_1}).$$

Therefore, the two cubical laws are compatible as long as $\widetilde{\phi}^{-1}(\widetilde{\mathcal{O}_{E_2}}) = \widetilde{\mathcal{O}_{E_1}}$. □

We also have a natural Galois action on cubical points, which is compatible with the cubical arithmetic. The cubical isomorphism from Proposition 3 and the cubical Galois action induce via Eq. (9) the corresponding biextension isomorphism and Galois action:

Corollary 3. *Let $g_{P,Q}$ be represented by $[\widetilde{P}, \widetilde{Q}; \widetilde{\mathcal{O}}_E, \widetilde{P+Q}]$. Then $\phi^* g_{P,Q}$ is represented by $[\widetilde{\phi}^{-1}(\widetilde{P}), \widetilde{\phi}^{-1}(\widetilde{Q}); \widetilde{\phi}^{-1}(\widetilde{\mathcal{O}}_E), \widetilde{\phi}^{-1}(\widetilde{P+Q})]$. And $\sigma \cdot g_{P,Q}$ is represented by $[\sigma(\widetilde{P}), \sigma(\widetilde{Q}); \sigma(\widetilde{\mathcal{O}}_E), \sigma(\widetilde{P+Q})]$.*

Example 1 (Level 2 cubical isomorphisms)**.** *In Section 2.1 the twisting isomorphisms $\phi : E' \to E$ are of the form $x \mapsto \xi^2 x$. Since we fix our level 2 neutral cubical point to be $\widetilde{\mathcal{O}}_E = (1, 0)$, we have that $\widetilde{\phi}(X, Z) = (X, Z/\xi^2)$ is a cubical isomorphism.*

Example 2 (Level 2 cubical automorphisms)**.** *On $E_2 : y^2 = x^3 + ax$ with $j(E_2) = 1728$, we have an automorphism $\sigma : (x, y) \mapsto (-x, iy)$ where $i^2 = -1$. This gives a level 2 cubical automorphism $\widetilde{\sigma} : (X, Z) \mapsto (X, -Z)$.*

On $E_1 : y^2 = x^3 + b$ with $j(E_1) = 0$, we have an automorphism $\sigma : (x, y) \mapsto (wx, y)$ where $w^3 = 1$. This gives a level 2 cubical automorphism $\widetilde{\sigma} : (X, Z) \mapsto (X, Z/w)$.

Remark 3. *In the remaining part of this paper, for ease of notations we will often* ***drop the tilde*** and use the notations X_P, Z_P (resp. $(X_1(P), Z_1(P))$ in level 2 (resp. level 1).

3 Main Results

In this section, we present a comprehensive framework to derive precise formulas for the ate pairing, optimal ate pairing, together with super-optimal ate pairing by leveraging the technique of biextension arithmetic. For each type of pairings, we first delineate the corresponding explicit formulas in level 1, and then operate on the Kummer line $K = E/\langle \pm 1 \rangle$ of an elliptic curve E in level 2 corresponding to the biextension $\mathrm{X}_{2(\mathcal{O}_E)}$ with sections (X, Z). Additionally, we provide illustrative examples.

3.1 Biextension for the Tate pairing

As a warm up, we first look at the Tate pairing and consider how to exploit twisting isomorphisms. Let $P \in E(\mathbb{F}_{p^k})[r]$ and $Q \in E(\mathbb{F}_{p^k})$. Then according to Section 2.2 the reduced Tate pairing is given by

$$e_r(P,Q) = f_{r,P}(Q)^{(p^k-1)/r} = f_{r,P}((Q+R)-(R))^{(p^k-1)/r}$$

for any rational point $R \in E(\mathbb{F}_{p^k})$. This definition can be mathematically reformulated using the framework of biextensions as follows.

Lemma 2. *For any $\mathbb{F}_{p^k}$-rational biextension function $g_{P,Q}$ above (P,Q), we have*

$$e_r(P,Q) = g_{[r]P,Q}^{(p^k-1)/r}.$$

Using the cubical arithmetic to compute the biextension exponentiation, we obtain

$$e_r(P,Q) = \left(\frac{Z_1([r]P+Q)}{Z_1(Q)Z_1([r]P)}\right)^{(q^k-1)/r}, \tag{12}$$

as long as we start with $\mathbb{F}_{p^k}$-rational (level 1) cubical points $P, Q, P+Q$, e.g. normalized to $Z_1(P) = Z_1(Q) = Z_1(P+Q) = 1$.

Proof. By Lemma 1, we have for any $R \in E(\mathbb{F}_{p^k})$,

$$e_r(P,Q) = e_r(-P,Q)^{-1} = \left(\frac{g_{[r]P,Q}}{g_{P,Q}^r}(R)\right)^{(p^k-1)/r}.$$

Now, if $g_{P,Q}$ is chosen to be $\mathbb{F}_{p^k}$-rational (e.g., the one normalized at $\mathcal{O}_E$), $g_{P,Q}^r(R)$ is killed by the final exponentiation. Since $[r]P = \mathcal{O}_E$, $g_{[r]P,Q}$ is a constant function. We then use Corollary 2 to obtain the cubical formulas. □

Now let $\phi : E' \to E$ be a twisting isomorphism, which is rational over $\mathbb{F}_{p^k}$. Let $P' = \phi^{-1}(P), Q' = \phi^{-1}(Q)$. The following lemma describes how to accomplish the reduced Tate pairing on the twist E'.

Lemma 3. *For any $\mathbb{F}_{p^k}$-rational biextension function $g_{P',Q'}$ above (P',Q'), we have:*

$$e_r(P,Q) = g_{[r]P',Q'}^{(p^k-1)/r},$$

In terms of cubical arithmetic, this can be restated as:

$$e_r(P,Q) = \left(\frac{Z_1([r]P'+Q')}{Z_1(Q')Z_1([r]P')}\right)^{(q^k-1)/r},$$

for any $\mathbb{F}_{p^k}$-rational cubical points $P', Q', P'+Q'$.

Proof. By Proposition 1, we have $g_{[r]P',Q'} = \phi^* g_{[r]P,Q}$, thus we can work on E' to compute the biextension exponentiation. Indeed, since ϕ is $\mathbb{F}_{p^k}$-rational, $g_{P',Q'}$ is $\mathbb{F}_{p^k}$-rational if and only if $g_{P,Q}$ is $\mathbb{F}_{p^k}$-rational.

Then to express the Tate pairing in term of the cubical points, we either apply Corollary 1 to $g_{[r]P',Q'}$, or we start with Eq. (12) and we use that $\widetilde{\phi}$ is a $\mathbb{F}_{p^k}$-rational cubical isomorphism by Proposition 3. □

There is a more intrinsic reformulation of Lemma 2 that does not depend on any $\mathbb{F}_{p^k}$-rational choice. Let $q = p^k$, and denote by $\pi_q \cdot g = \pi_q \circ g \circ \pi_q^{-1}$ the action on a function g by Galois conjugation, as described in Definition 3. By $[q-1]P = \mathcal{O}_E$ and $\pi_q(P) = P,\ \pi_q(Q) = Q$ we observe that

$$\begin{aligned}\operatorname{div}(\pi_q \cdot g_{P,Q}) &= (\pi_q(-P-Q)) + (\mathcal{O}_E) - (\pi_q(-P))) - (\pi_q(-Q)) \\ &= (-[q]P - Q) + (\mathcal{O}_E) - (-[q]P) - (-Q) \\ &= \operatorname{div}(g_{[q]P,Q}),\end{aligned}$$

hence $g_{[q]P,Q}$ and $\pi_q \cdot g_{P,Q}$ differ by a constant c. Furthermore, this constant does not depend on the choice of representative for $g_{P,Q}$, even non $\mathbb{F}_{p^k}$-rational. Hence, we can assume that $g_{P,Q}$ is $\mathbb{F}_{p^k}$-rational to determine c. Under this circumstance, we obtain

$$\pi_q \cdot g_{P,Q} = g_{P,Q}.$$

One can also prove that $g_{[q-1]P,Q}(\cdot + P)$ is a constant. On this basis, by Eq. (4) and the computation in the proof of Lemma 2 we derive that

$$\begin{aligned}g_{[q]P,Q}(R) &= c \cdot g_{P,Q}(R) \\ &= g_{[q-1]P,Q}(R) \star_1 g_{P,Q}(R) \\ &= g_{[q-1]P,Q}(R+P) \cdot g_{P,Q}(R) \\ &= g_{[q-1]P,Q}(R) \cdot g_{P,Q}(R) \\ &= g_{[r]P,Q}^{\star_1, \frac{q-1}{r}}(R) \cdot g_{P,Q}(R) \\ &= e_r(P,Q) \cdot g_{P,Q}.\end{aligned}$$

Consequently, we have $c = e_r(P,Q) = \frac{g_{[q]P,Q}}{\pi_q \cdot g_{P,Q}}$. In summary, for any biextension function $g_{P,Q}$, even non $\mathbb{F}_{p^k}$-rational, we have

$$e_r(P,Q) = \frac{g_{[q]P,Q}}{\pi_q \cdot g_{P,Q}}. \tag{13}$$

Remark 4. *One should be careful that for a general twisting isomorphism $\phi : E' \to E$ defined over an extension of $\mathbb{F}_{p^k}$, even if we start with a $\mathbb{F}_{p^k}$-rational biextension function $g_{P,Q}$, then $g_{P',Q'}$ may not be rational. If we compute the constant function $g_{[r]P',Q'}$ on E', starting with $g_{P',Q'}$ normalized at $\mathcal{O}_{E'}$ for ease of computation, then $g_{P',Q'} = \phi^* g_{P,Q}$ for some $g_{P,Q}$ that will not be normalized, nor even $\mathbb{F}_{p^k}$-rational. Hence in general, $g_{[r]P',Q'}(R')^{(p^k-1)/r}$ will not give the Tate pairing $e_r(P,Q)$.*

Instead, we need to use Eq. (13) to adjust the result to get the correct Tate pairing. More concretely, if $\phi : E' \to E$ is of the form $\phi(x, y) = (\xi^2 x, \xi^3 y)$, and we start with $g_{P,Q}$ normalized with respect to x/y, then $\phi^* g_{P,Q}$ is normalized with respect to $\phi^*(x/y) = \frac{1}{\xi} x'/y'$. So if we start with $g_{P',Q'}$ normalized with respect to x'/y', and we compute the constant function $c' = g_{[r]P',Q'}$, then we need to adjust c' by ξ^r to recover the constant function $c = g_{[r]P,Q} = c'/\xi^r$. Lemma 3 is the case where ϕ is $\mathbb{F}_{p^k}$-rational, so $\xi \in \mathbb{F}_{p^k}$ and ξ^r is killed by the final exponentiation.

A similar reasoning holds using the cubical arithmetic and the cubical isomorphism $\widetilde{\phi}$ when ϕ is defined over an extension of $\mathbb{F}_{p^k}$. But by the same computation as for biextensions above, if $Z_1(P) = 1$, then $Z_1(\widetilde{\phi}^{-1}(P)) = \xi \neq 1$. So conversely, if we want to use cubical arithmetic on E', and we start with normalized points $P', Q', P + Q'$ to speed up the cubical arithmetic, it means that going back to E we were doing cubical arithmetic with non normalized points, potentially even non rational cubical points. So we need to adjust by a suitable power of the conversion factor ξ in the end. It is only when ϕ is defined over $\mathbb{F}_{p^k}$ that this power of ξ lies in a strict subfield, so it will be killed by the final exponentiation anyway. In this paper we will only consider twisting isomorphisms that are rational over $\mathbb{F}_{p^k}$.

We can generalize Eq. (13) by relating the twisting correcting factor with the automorphism σ inducing the twist E', i.e. such that $\phi \circ \pi'_q \circ \phi^{-1} = \sigma \circ \pi_q$ where $\phi : E' \to E$ is the twisting isomorphism. Indeed, we have $\phi^{-1} \cdot g_{[q]P,Q} = g_{[q]P',Q'}$. However, $(\phi^{-1} \circ \pi_q) \cdot g_{P,Q}$ differs from $(\pi'_q \circ \phi^{-1}) \cdot g_{P,Q}$ in general. Therefore, unraveling the formulas we obtain

$$(\pi'_q \circ \phi^{-1}) \cdot g_{P,Q} = ({\sigma'}^{-1} \circ \phi \circ \pi_q) \cdot g_{P,Q} = (\phi^{-1} \circ \pi_q \circ \sigma) \cdot g_{P,Q}, \tag{14}$$

where $\sigma' = \phi^{-1} \circ \sigma^{-1} \circ \phi$, i.e., $\phi^{-1} \circ \pi_q \circ \phi = \sigma' \circ \pi'_q$. In particular,

$$(\phi^{-1} \circ \pi_q) \cdot g_{P,Q} = (\sigma' \circ \pi'_q \circ \phi^{-1}) \cdot g_{P,Q}.$$

It follows that

$$e_r(P,Q) = \left(\frac{g_{[q]P,Q} \circ \phi}{(\pi_q \cdot g_{P,Q}) \circ \phi} \right)^{\frac{p^k-1}{r}} = \left(\frac{\phi^{-1}(g_{[q]P,Q})}{(\phi^{-1} \circ \pi_q) \cdot g_{P,Q}} \right)^{\frac{p^k-1}{r}} = \left(\frac{g_{[q]P',Q'}}{(\sigma' \circ \pi'_q) \cdot g_{P',Q'}} \right)^{\frac{p^k-1}{r}}$$

for *any* twist isomorphism $\phi : E' \to E$, even non $\mathbb{F}_{p^k}$-rational, and *any* biextension function $g_{P',Q'}$.

3.2 Biextension for Ate Pairing

As discussed in Section 2.1, the ate pairing is a variant of the Tate pairing that employs the p-power Frobenius endomorphism π to reduce the length of the Miller loop. Using the same notation, the reduced ate pairing on E is defined as

$$a_\lambda(P,Q) = (f_{\lambda,Q}(P))^{\frac{p^k-1}{r}}$$

where $P \in \mathbb{G}_1$ and $Q \in \mathbb{G}_2$. Given that $\lambda \equiv p \mod r$, for instance $\lambda = t - 1$, with t the trace of the Frobenius endomorphism.

Working in level 1 biextension and cubical arithmetic, taking the normalized biextension function $g_{P,Q}$, and setting $Z_1(P) = Z_1(Q) = Z_1(P + Q) = 1$, it follows from Lemma 1 that the the reduced ate pairing can (up to a sign) be expressed as

$$a_\lambda(P, Q) = \left(\frac{g_{[\lambda]Q,P}}{g^p_{Q,P}}(\mathcal{O}_E) \right)^{\frac{p^k-1}{r}} = \left(\frac{Z_1([\lambda]Q + P)}{Z_1([\lambda]Q)} \right)^{\frac{p^k-1}{r}}.$$

We remark that if we take $\lambda = p$ instead of $\lambda = t - 1$, we have a more intrinsic characterization of the reduced ate pairing, similar to Eq. (13): $e_\lambda(P, Q) = \frac{g_{[\lambda]Q,P}}{\pi_p \cdot g_{Q,P}}$. Indeed, the functions in the numerator and denominator have the same divisor, thus their quotient is constant. It is readily seen that this quotient does not depend on the choice of $g_{Q,P}$ even non $\mathbb{F}_{p^k}$-rational, hence we do not need the final exponentiation.

In practical applications, most of the curves utilized in pairing-based cryptography admit twists. Therefore, it is essential to employ the twisting isomorphism ϕ to enhance the efficiency of ate pairing. According to [20], for elliptic curves admitting twists, the pairing subgroup $\mathbb{G}_2$ can be represented as

$$\mathbb{G}_2 \cong E'(\mathbb{F}_{p^{k/d}})[r].$$

We now elucidate how to exploit the technique of twists to compute the ate pairing on the Kummer line $K = E/\langle \pm 1 \rangle$ via biextension. Let $P \in \mathbb{G}_1$ and $Q' \in E'(\mathbb{F}_{p^{k/d}})[r]$ such that $Q = \phi(Q') \in \mathbb{G}_2$, where $\phi : E' \to E$ is the degree-d twisting isomorphism. Let $P' = \phi^{-1}(P)$.

Lemma 4. *With the notations as above, if we let σ' be the automorphism on E' induced by ϕ, i.e. $\phi^{-1} \circ \pi_p \circ \phi = \sigma' \circ \pi'_p$, then the ate pairing is given by*

$$\begin{aligned} a_\lambda(P, Q) &= \left(\frac{g_{[\lambda]Q,P}}{\pi_p \cdot g_{Q,P}}(\mathcal{O}_E) \right)^{(q^k-1)/r} \\ &= \left(\frac{g_{[\lambda]Q',P'}}{(\sigma' \circ \pi'_p) \cdot g_{Q',P'}}(\mathcal{O}_{E'}) \right)^{(q^k-1)/r} \end{aligned}$$

for any biextension $g_{Q,P}$ (resp. $g_{Q',P'}$) that is rational over $\mathbb{F}_{p^k}$ (in particular we can take the normalized biextension functions).

Proof. Since $(g_{[\lambda]Q,P}/\pi_p \cdot g_{Q,P})$ is a constant function, we can evaluate it on $\mathcal{O}_E$ to determine its value, which gives the first equality. The second equality follows by applying the action of ϕ on the quotient, and by Eq. (14) we have

$$\phi^{-1} \circ \pi_p = \sigma' \circ \pi'_p \circ \phi^{-1}.$$

(An alternative proof is that we have just seen that the result is invariant under the choice of a $\mathbb{F}_{p^k}$-rational representative of $g_{Q,P}$, and although ϕ^* is not rational over $\mathbb{F}_p$, it is a rational biextension isomorphism over $\mathbb{F}_{p^k}$.) □

We now switch to level 2 cubical arithmetic because the associated formulas are more convenient, hence compute the square of the ate pairing.

Theorem 1. *With the aforementioned notations, then the reduced ate pairing on $K = E/\langle \pm 1 \rangle$ corresponding to the biextension $X_{2(\mathcal{O}_E)}$ with sections (X, Z) can be computed as*

$$a_{b,\lambda}(P,Q) = a_\lambda(P,Q)^2 = Z_{[\lambda]Q'+P'}^{\frac{p^k-1}{r}}.$$

where $P' = \phi^{-1}(P)$ and the twisted cubical points are normalized via $Z_{P'} = Z_{Q'} = Z_{P'+Q'} = 1$.

Proof. The level 2 ate pairing is given by

$$a_b(P,Q) = \left(\frac{Z_{[\lambda]Q+P}}{Z_{[\lambda]Q}}\right)^{\frac{p^k-1}{r}}.$$

where $Z_P = Z_Q = Z_{P+Q} = 1$. We now use the cubical isomorphism of Example 1. (An alternative proof would be to start with the second equality of Lemma 4).

Then $Z_{\phi^{-1}(P)} = Z_P/\xi^2$, so $\phi^{-1}(P)$ is not normalized, and similarly for $\phi^{-1}(Q), \phi^{-1}(P+Q)$. From Remark 4, if we start with normalized points for $P', Q', P'+Q'$, the resulting coordinate $Z_{[\lambda]Q'+P'}$ is off compared to $Z_{[\lambda]Q+P}$ by some power of ξ^2, which is killed by the final exponentiation. Consequently, the numerator is given by

$$Z_{[\lambda]Q+P}^{(q^k-1)/r} = \phi_*(Z_{[\lambda]Q'+P'})^{(q^k-1)/r} = Z_{[\lambda]Q'+P'}^{(q^k-1)/r},$$

since $\phi_* Z = \xi^2 Z$.

The same argument for the denominator shows that $Z_{[\lambda]Q}^{(q^k-1)/r} = Z_{[\lambda]Q'}^{(q^k-1)/r}$. Since Q' lies in the subfield $\mathbb{F}_{p^{k/d}}$, this denominator vanishes in the final exponentiation. □

By utilizing the twisting isomorphism, part of the computations can be performed in the subfield $\mathbb{F}_{p^{k/d}}$. Taking $\lambda = t-1$, the coordinate $Z_{[t-1]Q'+P'}$ can be obtained via the cubical ladder algorithm. The detailed computational procedures for the ate pairing through biextension are presented in Section 4.1. For some specific families of pairing-friendly curves, such as BN12 and BLS12, the number of basic Miller iterations of ate pairings is exactly $\log_2(r)/\varphi(k)$. In other words, for these curves the ate pairing itself is optimal. We provide the following example for illustration.

Example 3 (BLS12 Family)**.** *The BLS12 family, with embedding degree $k = 12$ and CM-discriminant $D = 3$, is popular in pairing-based cryptography. Notable pairing-friendly curves such as BLS12-377, BLS12-381, and BLS12-446 have been employed in numerous cryptographic schemes. The parameters r, t, and p are parametrized as*

follows

$$\begin{aligned} r(z) &= z^4 - z^2 + 1, \\ t(z) &= z + 1, \\ p(z) &= \frac{(z^2 - 2z + 1)(z^4 - z^2 + 1)}{3} + z. \end{aligned}$$

It is worth noting that $t(z) - 1 = z$*, which is close to* $r(z)^{1/4} = r(z)^{1/\varphi(k)}$*. Therefore, the ate pairings on these curves are indeed optimal ate pairings. Additionally, there exists a sextic twist* E' *for a BLS12 curve* E*. As mentioned in Section 2.3, the twisting isomorphism is* $\phi : E' \to E$*,* $(x, y) \mapsto (\zeta^{\frac{1}{3}}x, \zeta^{\frac{1}{2}}y)$ *with* $\zeta \in \mathbb{F}_{p^2}^* \mod (\mathbb{F}_{p^2}^*)^6$*. By Theorem 1, the ate pairing on* $E/\langle \pm 1 \rangle$ *via biextension can be computed as*

$$a_b(P, Q) = Z_{[z]Q'+P'}^{\frac{p^{12}-1}{r}}.$$

3.3 Biextension for optimal ate pairing

In this subsection, we derive the formulas for optimal ate pairings through biextension by utilizing the technique of twists. From Section 2.2, we consider the multiple $\lambda = mr = \sum_{i=0}^{l} c_i p^i$, where the short vector $(c_0, c_1, \ldots, c_l)$ satisfies $|c_i| \approx r^{\frac{1}{\varphi(k)}}$. Using the formula for the optimal ate pairing in Eq. (3) on $\mathbb{G}_2 \times \mathbb{G}_1$, Lemma 1 and Eq. (4) show that the biextension interpretation of the optimal ate pairing can be expressed as

$$\begin{aligned} opt(P, Q) &= \left(g_{[c_0]Q,P} \star_1 \pi_p \cdot g_{[c_1]Q,P} \star_1 \cdots \star_1 \pi_p^l \cdot g_{[c_l]Q,P}\right)^{\frac{p^k-1}{r}} (\mathcal{O}_E) \\ &= \left(\prod_{\star_1, i=0}^{l} \pi_p^i \cdot g_{[c_i]Q,P}\right) (\mathcal{O}_E)^{\frac{p^k-1}{r}}, \end{aligned}$$

for any biextension function $g_{P,Q}$ rational over $\mathbb{F}_{p^k}$. See also [30, Section 3.4] for the associated monodromy interpretation: in the non-reduced Tate pairing, we compute the constant function $g_{[r]Q,P} = \prod_{\star_1, i=0}^{l} g_{[c_i]Q,P}^{\star_1, p^i}$. In the optimal ate pairing, we replace $g_{[c_i]Q,P}^{\star_1, p^i}$ by $\pi^i \cdot g_{[c_i]Q,P}$, which differs from the above function by some ate pairing by Section 3.2.

Using the cubical representation of biextension functions, we obtain that for normalized level 1 cubical points,

$$opt(P, Q) = \left(\frac{Z_1(\sum_{i=0}^{l} \pi^i([c_i]Q) + P)}{Z_1(\sum_{i=0}^{l} \pi^i([c_i]Q))}\right)^{\frac{p^k-1}{r}}.$$

Similar to the ate pairing, the technique of twists can also be employed to enhance computational efficiency.

Lemma 5. *With the same notations as Lemma 4, we have*

$$opt(P,Q) = \left(\prod_{\star_1,i=0}^{l} \pi_p^i \cdot g_{[c_i]Q,P}\right)^{\frac{p^k-1}{r}}$$
$$= \left(\prod_{\star_1,i=0}^{l} \left(\sigma'^i \circ \pi_p'^i\right) \cdot g_{[c_i]Q',P'}\right)^{\frac{p^k-1}{r}}$$

for any $\mathbb{F}_{p^k}$-rational biextension function $g_{P,Q}$ and $g_{P',Q'}$ respectively.

Proof. We use the same proof as Lemma 4. Since $\prod_{\star_1,i=0}^{l} \pi_p^i \cdot g_{[c_i]Q,P}$ is a constant function, we can evaluate it on $\mathcal{O}_E$ to recover its value, which gives the first equality. Also it is easy to see that the value does not depend on the choice of $\mathbb{F}_{p^k}$-rational representative of $g_{P,Q}$ because of the final exponentiation. The second equality follows by applying ϕ to this constant function g. □

We now present the following theorem to illustrate the formulas for optimal ate pairings on Kummer lines through level 2 cubical arithmetic by exploiting twists.

Theorem 2. *Using the above notations, let $P \in \mathbb{G}_1$ and $Q' \in E'(\mathbb{F}_{p^{k/d}})[r]$ such that $Q = \phi(Q') \in \mathbb{G}_2$. The optimal ate pairing on $K = E/\langle \pm 1 \rangle$ corresponding to the biextension $X_{2(\mathcal{O}_E)}$ can be computed as*

$$opt_b(P,Q) = opt(P,Q)^2 = \left(Z_{\sum_{i=0}^{l}(\sigma'^i \circ \pi_p'^i)([c_i]Q')+P'}\right)^{\frac{p^k-1}{r}}.$$

Proof. We have $opt_b(P,Q) = \left(\frac{Z_{\sum_{i=0}^{l} \pi_p{}^i([c_i]Q)+P}}{Z_{\sum_{i=0}^{l} \pi_p^i([c_i]Q)}}\right)^{\frac{p^k-1}{r}}$. We now apply the cubical isomorphism ϕ^{-1}, using Proposition 3 and Corollary 3, to obtain:

$$opt_b(P,Q) = \left(\frac{Z_{\sum_{i=0}^{l}(\sigma'^i \circ \pi_p'^i)([c_i]Q')+P'}}{Z_{\sum_{i=0}^{l}(\sigma'^i \circ \pi_p'^i)([c_i]Q')}}\right)^{\frac{p^k-1}{r}}.$$

Indeed, by the same argument as in the proof of Theorem 1, the correcting factor ξ^2 and the denominator vanish in the final exponentiation. (An alternative proof is to employ Lemma 5). □

To simplify the notation, when working on E' we will denote by π (or π_p) the "corrected" Frobenius $\sigma' \circ \pi_p'$ where π_p' is the standard Frobenius on E'. The notation is chosen such that $(\pi \circ \phi) \cdot g = (\phi \circ \pi) \cdot g$. According to the above proof and Corollary 3, we require the computation of the following coordinates

$$Z_{\pi^i([c_i]Q'+P')} \quad \text{and} \quad Z_{\pi^i([c_i]Q')}, \quad i = 0, \ldots, l,$$

which can be achieved through the following steps

1. Compute $Z_{[c_i]Q'}$ and $Z_{[c_i]Q'+P'}$ for $i = 0, \ldots, l$ using the cubical or double-and-add ladder algorithm.
2. Apply the morphisms π^i separately to the points $[c_i]Q'$ and $[c_i]Q' + P$ to obtain $Z_{\pi^i([c_i]Q')}$ and $Z_{\pi^i([c_i]Q'+P')}$.
3. Compute $Z_{\sum_{i=0}^{l}\pi^i([c_i]Q')}$ and $Z_{\sum_{i=0}^{l}\pi^i([c_i]Q')+P'}$ from the points $Z_{\pi^i([c_i]Q')}$ and $Z_{\pi^i([c_i]Q'+P')}$ using the three-way addition algorithm [30] combined with Remark 2.

The most computationally expensive step is the calculation of $Z_{[c_i]Q'}$ and $Z_{[c_i]Q'+P'}$. By employing the technique of twists, part of the computation can be performed over the subfield $\mathbb{F}_{p^{k/d}}$, compared to the original approach in [30]. Detailed algorithms and cost analysis are provided in Section 4.2. In the following, we present the AFG16 family as a concrete example.

Example 4 (AFG16 Family)**.** *The AFG16 family, with embedding degree $k = 16$ and CM-discriminant $D = 1$, is known for efficient pairing computation and hashing, making it competitive in pairing-based cryptography. The parametrized polynomials $r(z)$, $t(z)$, and $p(z)$ are given by:*

$$\begin{aligned} r(z) &= \Phi_{16}(z) = z^8 + 1, \\ t(z) &= r(z) + z^5 + 1 = z^8 + z^5 + 2, \\ p(z) &= \frac{z^{16} + 2z^{13} + z^{10} + 5z^8 + 6z^5 + z^2 + 4}{4}. \end{aligned}$$

There exists a quartic twist E' for an AFG16 curve E. From Section 2.3, the twisting isomorphism is defined as $\phi : E' \to E$, $(x, y) \mapsto (D^{\frac{1}{2}}x, D^{\frac{3}{4}}y)$ with $D \in \mathbb{F}_{p^4}^*$ mod $(\mathbb{F}_{p^4}^*)^4$. Additionally, it holds that $z + p^5 \equiv 0 \mod r$ on AFG16. By Theorem 2, the optimal ate pairing on AFG16 can be computed through biextension as:

$$\begin{aligned} opt_b(P, Q) &= \left(g_{[z]Q',P'} \star_1 g_{\pi^5(Q'),P'}\right)^{\frac{p^{16}-1}{r}} \\ &= \left(Z_{[z]Q'+\pi^5(Q')+P'}\right)^{\frac{p^{16}-1}{r}}. \end{aligned}$$

By Eq. (6), we have

$$g_{[z]Q',P'} \star_1 g_{\pi^5(Q'),P'} = g_{[z]Q',P'}(\cdot) \cdot g^{p^5}_{Q',P'}(\cdot) \cdot v^2_{[p^5]Q'}(P').$$

Since the function $g_{Q',P'}$ is normalized, and $v^2_{[p^5]Q'}(P')$ can be killed by the final exponentiation, it suffices to compute

$$opt_b(P, Q) = g^{\frac{p^{16}-1}{r}}_{[z]Q',P'} = Z^{\frac{p^{16}-1}{r}}_{[z]Q'+P'}.$$

3.4 Biextension for super-optimal ate pairing

The super-optimal pairings are meticulously constructed on specific families of pairing-friendly curves by using GLV-automorphisms. To enhance the efficiency, automorphisms are frequently employed to derive the formulas of super-optimal pairings on curves with j-invariants $j = 0$ or 1728. In this subsection, we primarily focus on deriving the formulas for super-optimal pairings on such GLV-curves endowed with efficiently-computable automorphisms, including curves that admit twists and those that with the lack of twists, through the framework of biextensions.

In Section 3.3 for the optimal pairing, if $r = \sum_i c_i p^i$, we were using the fact that $\pi_p(Q) = [p]Q$ and $\pi_p(P) = P$, to replace in the Tate pairing biextension exponentiation $g_{Q,P}^{\star_1,r} = \prod_{\star_1,i} g_{[c_i p^i]Q,P}$ the biextension function $g_{[c_i p^i]Q,P} = g_{[c_i]Q,P}^{\star_1,p^i}$ by $\pi_p^i \cdot g_{[c_i]Q,P}$. Indeed, both functions have the same divisor, hence differ by a constant (given by some ate pairings). On several specific pairing-friendly curves E admitting extra automorphisms σ, we can combine these automorphisms with the power of Frobenius endomorphism to determine $\tau = \pi^j \circ \sigma$ such that $\tau(Q) = [z]Q$, where z is the parametrized seed of the families of pairing-friendly curves. Then we can use the same strategy as for the optimal ate pairing, writing $r = \sum c_i z^i$ and replacing $g_{[c_i]Q,P}^{\star_1,z^i}$ by $\tau^i \cdot g_{[c_i]Q,\tau^{-i}(P)}$, which has the same divisor, hence differ by a constant. Here we just need to be careful that τ does not fix P, and we need to normalize the $g_{Q,\tau^{-i}(P)}$ appropriately.

Our objective is to derive the super-optimal pairings on the following two types of GLV-curves, E_1 and E_2, as described in Section 2.1:

$$\begin{aligned} E_1 &: y^2 = x^3 + b, \quad j(E_1) = 0, \\ E_2 &: y^2 = x^3 + ax, \quad j(E_2) = 1728. \end{aligned}$$

We will denote by σ the extra automorphisms on these curves (See Section 2.1 for more details).

According to [10, 9], in practice E_1 (resp. E_2) precisely corresponds to a pairing-friendly curve in the completed family Cyclo (6.6) (resp. Cyclos (6.2), (6.3), (6.4), or (6.5)) [15] parametrized by the polynomials $p(z), t(z), r(z)$. Additionally, as noted in [9], for any $Q \in \mathbb{G}_2 \subseteq E_i(\mathbb{F}_{p^k})$, there exists a positive integer j $(1 \leq j < k)$ such that

$$\tau(Q) = \pi^j \circ \sigma(Q) = [z]Q.$$

If the embedding degree k satisfies $\mathrm{ord}(\sigma) \nmid k$, it enables us to reduce the number of Miller iterations to approximately $\log_2(r)/2\varphi(k)$ [21, 9] and construct the super-optimal ate pairings on the corresponding curves E_i $(i = 1, 2)$.

For simplicity, we first look at the case where $j(E_1) = 0$. Let ζ_k denote the k-th primitive roots of unity. By the characteristic equations of σ, we observe that it corresponds to $\frac{-1\pm\sqrt{-3}}{2}$ in $\mathrm{End}(E_1)$. Besides, π acts like $[p] = \zeta_k$ a k-th root of unity on $\mathbb{G}_2$. Consequently, we obtain on this subgroup, it holds that

$$\begin{aligned}\tau^2 + \tau p^j + p^{2j} &= \tau^2 + \tau \cdot \zeta^j + \zeta^{2j} \\ &= \zeta^{2j} \cdot \left(\left(\frac{-1 \pm \sqrt{-3}}{2} \right)^2 + \frac{-1 \pm \sqrt{-3}}{2} + 1 \right) \\ &= 0.\end{aligned}$$

By $\tau(Q) = [z]Q$, we have

$$z^2 + zp^j + p^{2j} \equiv 0 \mod r. \tag{15}$$

By Eq. (3), we can derive a super-optimal ate pairing

$$sopt(P,Q) = \left(f_{z^2,Q}(P) \cdot f_{z,Q}^{p^j}(P) \cdot \ell_{[p^{2j}]Q,[zp^j]Q}(P) \right)^{\frac{p^k-1}{r}} \tag{16}$$

$$= \left(f_{z,Q}^{z+p^j}(P) \cdot f_{z,Q}^{p^j}(\sigma^{-1}(P)) \cdot \ell_{[p^{2j}]Q,[zp^j]Q}(P) \right)^{\frac{p^k-1}{r}}. \tag{17}$$

To leverage the biextension arithmetic, we need the following lemma.

Lemma 6. *For any $\mathbb{F}_{p^k}$-rational biextension function $g_{Q,P}$, we have:*

$$sopt(P,Q) = \left(\tau \cdot g_{[z]Q,\sigma^{-1}(P)} \star_1 \pi^j \cdot g_{[z]Q,P} \star_1 \pi^{2j} \cdot g_{Q,P} \right)^{\frac{p^k-1}{r}},$$

where $g_{Q,\sigma^{-1}(P)} = \tau^{-1} \cdot g_{[z]Q,P}$.

Proof. Utilizing the formulas $[z^2 + zp^j + p^{2j}]Q = \mathcal{O}_E$ for the optimal pairing , we have

$$sopt(P,Q) = \left(g_{[z^2]Q,P} \star_1 \pi^j \cdot g_{[z]Q,P} \star_1 \pi^{2j} \cdot g_{Q,P} \right)^{\frac{p^k-1}{r}}$$

for any $\mathbb{F}_{p^k}$-rational function $g_{Q,P}$.

Now, if we take $g_{Q,\sigma^{-1}(P)}$ normalized so that $\tau \cdot g_{Q,\sigma^{-1}(P)} = g_{[z]Q,P}$, then by the compatibility of the action with the biextension arithmetic, we have:

$$\tau \cdot g_{Q,\sigma^{-1}(P)}^{\star_1,z} = (\tau \cdot g_{Q,\sigma^{-1}(P)})^{\star_1,z} = g_{[z]Q,P}^{\star_1,z} = g_{[z^2]Q,P}.$$

Plugging this in the equation above, we obtain:

$$sopt(P,Q) = \left(\tau \cdot g_{[z]Q,\sigma^{-1}(P)} \star_1 \pi^j \cdot g_{[z]Q,P} \star_1 \pi^{2j} \cdot g_{Q,P} \right)^{\frac{p^k-1}{r}}.$$

□

To compute the super-optimal pairing by the formulas in Lemma 6, we start with two normalized functions $g_{Q,P}$ and $g_{Q,\sigma^{-1}(P)}$, hence $\tau^{-1} \cdot g_{[z]Q,P}$ and $g_{Q,\sigma^{-1}(P)}$ differ by a constant λ. The explicit formula is illustrated in Lemma 7.

Lemma 7. *With the normalized biextension functions $g_{Q,\sigma^{-1}(P)}$ and $g_{Q,P}$, we have:*

$$sopt(P,Q) = \left(g^{p^j}_{[z]Q,\sigma^{-1}(P)}(\mathcal{O}_E) g^{p^j+z}_{[z]Q,P}(\mathcal{O}_E) \frac{g_{[z^2]Q,\pi^j([z]Q)}((P)-(\mathcal{O}_E))}{v_{\pi^{2j}(Q)}(P)} \right)^{(p^k-1)/r} \tag{18}$$

$$= \left(\frac{g^{p^j}_{[z]Q,\sigma^{-1}(P)}(\mathcal{O}_E) g^{p^j+z}_{[z]Q,P}(\mathcal{O}_E)}{\ell_{-\pi^{2j}(Q),-\pi^j([z]Q)}(P)} \right)^{(p^k-1)/r} \tag{19}$$

Proof. Write $\tau^{-1} \cdot g_{[z]Q,P} = \lambda g_{Q,\sigma^{-1}(P)}$ for some unknown constant λ. If we start with these normalized functions, then we have

$$sopt(P,Q) = \left(\lambda^{zp^j} \tau \cdot g_{[z]Q,\sigma^{-1}(P)} \star_1 \pi^j \cdot g_{[z]Q,P} \star_1 \pi^{2j} \cdot g_{Q,P} \right)^{(p^k-1)/r}.$$

The equation $\lambda^{p^j} \tau \cdot g_{Q,\sigma^{-1}(P)} = g_{[z]Q,P}$ allows us to recover λ. If we evaluate this equation on $\mathcal{O}_E$ (using an extended value), since $g_{Q,\sigma^{-1}(P)}$ is normalized we deduce that

$$\lambda^{p^j} = g_{[z]Q,P}(\mathcal{O}_E) / g_{Q,\sigma^{-1}(P)}(\mathcal{O}_E)^{p^j} = g_{[z]Q,P}(\mathcal{O}_E).$$

Substituting λ^{p^j} in the equation above and exploiting the biextension law, we obtain:

$$sopt(P,Q) = \left(g^{p^j}_{[z]Q,\sigma^{-1}(P)}(\mathcal{O}_E) \cdot g^{z+p^j}_{[z]Q,P}(\mathcal{O}_E) \cdot \frac{(g_{[z^2]Q,[p^j z]Q} \cdot g_{[z^2+p^j z]Q,[p^{2j}]Q})(P)}{(g_{[z^2]Q,[p^j z]Q} \cdot g_{[z^2+p^j z]Q,[p^{2j}]Q})(\mathcal{O}_E)} \right)^{(p^k-1)/r}.$$

Finally, since $[p^{2j} + p^j z + z^2]Q = \mathcal{O}_E$ and $g_{[p^j+z^2]Q,[p^{2j}]Q}((P) - (\mathcal{O}_E)) = 1/v_{[p^{2j}]Q}(P)$, we have

$$\begin{aligned}(g_{[z^2]Q,[p^j z]Q} \cdot g_{[p^j z+z^2]Q,[p^{2j}]Q})((P)-(\mathcal{O}_E))) &= \frac{v_{[z^2+p^j z]Q}(P)}{\ell_{[-p^j z]Q,[-z^2]Q}(P)} \cdot \frac{1}{v_{[p^{2j}]Q}(P)} \\ &= \frac{1}{\ell_{[-p^{2j}]Q,[-p^j z]Q}(P)}.\end{aligned}$$

Plugging it into the equation above, we complete the proof. □

We remark that by Lemma 1, we can rewrite Eq. (19) as

$$sopt(P,Q) = 1/\left(f^{z+p^j}_{z,-Q}(P) \cdot f^{p^j}_{z,-Q}(\sigma^{-1}(P)) \cdot \ell_{[-p^{2j}]Q,[-zp^j]Q}(P) \right)^{\frac{p^k-1}{r}},$$

which gives us back Eq. (17) using the fact that $sopt(P,Q) = 1/sopt(P,-Q)$.

In practice, we can use the cubical arithmetic to compute the super-optimal ate pairing:

$$sopt(P,Q) = \left(\frac{Z_1([z^2]Q + \pi^j([z]Q) + \pi^{2j}(Q) + P)}{Z_1([z^2]Q + \pi^j([z]Q) + \pi^{2j}(Q))}\right)^{\frac{p^k-1}{r}} \tag{20}$$

$$= \left(\frac{Z_1(\tau([z]Q) + \pi^j([z]Q) + \pi^{2j}(Q) + P)}{Z_1(\tau([z]Q) + \pi^j([z]Q) + \pi^{2j}(Q))}\right)^{\frac{p^k-1}{r}} \tag{21}$$

More precisely, in the formula above we start with arbitrary $\mathbb{F}_{p^k}$-rational cubical points for $P, Q, P+Q$. Then we compute $[z]Q, [z]Q+P$ using the cubical arithmetic, apply τ^{-1} to $[z]Q$, $[z]Q+P$ and P to get $Q_0 = \lambda_0 Q$, $Q_0+\sigma^{-1}P$ and $\sigma^{-1}P$ respectively, which we use to compute $[z]Q_0$, $[z]Q_0 + \sigma^{-1}P$. We then apply τ to these two points to get $[z^2]Q, [z^2]Q+P$. We also apply π^j to $[z]Q, [z]Q+P$ to get $[p^j z]Q, [p^j z]Q+P$, and π^{2j} to $Q, Q+P$ to get $[p^{2j}]Q, [p^{2j}]Q+P$. We now use arbitrary choices to get $[z]Q+[p^j z]Q+[p^{2j}]Q$ and threeway additions from these choices to get $[z]Q+[p^j z]Q+[p^{2j}]Q+P$. The second point is equal to P up to some projective factor which is the numerator of Eq. (21) and the first point is equal to $\mathcal{O}_E$ up a some projective factor which is the denominator of Eq. (21).

Similar to the proof in Lemma 7, we can start with normalized cubical points to simplify the formulas, by plugging the cubical formulas in Eq. (19).

Lemma 8. *Assume that we start with normalized cubical points $P, Q, Q+P, \sigma^{-1}(P), Q+\sigma^{-1}(P)$ and compute $[z]Q, [z]Q+P, [z]Q+\sigma^{-1}(P)$. In addition, we compute $\tau([z]Q) + \pi^j([z]Q) + P$ via a three way addition between $\tau([z]Q), \pi^j([z]Q), P, \pi^j([z]Q+P), \tau([z]Q+\sigma^{-1}P), -\pi^{2j}Q$. Then it yields that*

$$sopt(P,Q) = \left(\frac{Z_1([z]Q+P)^z \cdot Z_1(\tau([z]Q) + \pi^j([z]Q) + P)}{Z_1([z]Q)^z \cdot v_{\pi^{2j}(Q)}(P)}\right)^{(p^k-1)/r}. \tag{22}$$

Proof. By Lemma 7, we have

$$sopt(P,Q) = \left(g^{p^j}_{[z]Q,\sigma^{-1}(P)}(\mathcal{O}_E) \cdot g^{p^j+z}_{[z]Q,P}(\mathcal{O}_E) \cdot \frac{g_{[z^2]Q,\pi^j([z]Q)}((P)-(\mathcal{O}_E)))}{v_{\pi^{2j}(Q)}(P)}\right)^{(p^k-1)/r}.$$

Since the cubical point $\tau([z]Q)+\pi^j([z]Q)+P$ is computed by the three-way addition. By the cubical arithmetic and Eq. (10), we obtain

$$\frac{g_{[z^2]Q,[p^j z]Q}(P)}{g_{[z^2]Q,[p^j z]Q}(\mathcal{O}_E)} = \frac{Z_1(\tau([z]Q) + \pi^j([z]Q) + P)Z_1(P)Z_1(\tau([z]Q))Z_1(\pi^j([z]Q))}{Z_1(\tau([z]Q) + P)Z_1(\pi^j([z]Q) + P)Z_1(\tau([z]Q) + \pi^j([z]Q))Z_1(\mathcal{O}_E)}.$$

Note that we can always take $Z_1(P) = Z_1(\mathcal{O}_E) = 1$ (by the extend value). Moreover, it follows from Example 2 that

$$Z_1(\tau([z]Q)) = Z_1([z]Q^{p^j})/w,\ Z_1(\pi^j([2]Q)) = Z_1([z]Q)^{p^j},$$

$$Z_1(\tau([z]Q)+\pi^j([2]Q)) = Z_1(-\pi^{2j}(Q)) = -Z_1(Q)^{p^{2j}} = -1.$$

Substituting these relationships into the equation above, we have

$$\frac{g_{[z^2]Q,[p^j z]Q}(P)}{g_{[z^2]Q,[p^j z]Q}(\mathcal{O}_E)} = -\frac{Z_1(\tau([z]Q)+\pi^j([z]Q)+P)Z_1([z]Q)^{2p^j}}{wZ_1(\tau([z]Q)+P)Z_1(\pi^j([z]Q)+P)}.$$

Plugging this equation in Eq. (19), and by the fact that $-w$ vanishes in the final exponentiation, we complete the proof of this lemma. □

The formulas for the super-optimal ate pairings on E_2 with $j(E_2)=1728$ via biextension can be derived similarly. We just state the results in the following proposition for simplicity.

Proposition 4. *On E_2 with $j(E_2)=1728$, we have $z^2+p^{2j}\equiv 0 \mod r$. It follows that the super-optimal pairing is given by*

$$\begin{aligned} sopt(P,Q) &= \left(g_{[z^2]Q,P} \star_1 \pi^{2j}\cdot g_{Q,P}\right)^{\frac{p^k-1}{r}} \\ &= \left(\tau\cdot g_{[z]Q,\sigma^{-1}(P)} \star_1 \pi^{2j}\cdot g_{Q,P}\right)^{\frac{p^k-1}{r}}. \end{aligned}$$

for any $\mathbb{F}_{p^k}$-rational function $g_{Q,P}$, where $g_{Q,\sigma^{-1}(P)} = \tau^{-1}\cdot g_{[z]Q,P}$. Starting with normalized biextension functions $g_{Q,P}$ and $g_{Q,\sigma^{-1}(P)}$, we have

$$sopt(P,Q) = \left(g^{p^j}_{[z]Q,\sigma^{-1}(P)}(\mathcal{O}_E)g_{[z]Q,P}(\mathcal{O}_E)^z / v_{\pi^{2j}(Q)}(P)\right)^{\frac{p^k-1}{r}}.$$

Using the cubical arithmetic, we get

$$sopt(P,Q) = \left(\frac{Z_1(\tau([z]Q)+\pi^{2j}(Q)+P)}{Z_1(\tau([z]Q)+\pi^{2j}(Q))}\right)^{\frac{p^k-1}{r}}.$$

Starting with normalized cubical points for $P, Q, Q+P, \sigma^{-1}(P), Q+\sigma^{-1}(P)$, this can be simplified to

$$\begin{aligned} sopt(P,Q) &= \left(\left(\frac{Z_1([z]Q+\sigma^{-1}P)}{Z_1([z]Q)}\right)^{p^j}\cdot\left(\frac{Z_1([z]Q+P)}{Z_1([z]Q)}\right)^{z}\cdot\frac{1}{v_{\pi^{2j}(Q)}(P)}\right)^{(p^k-1)/r} \\ &= \left(\frac{Z_1([z]Q+\sigma^{-1}P)^{p^j}Z_1([z]Q+P)^z}{Z_1([z]Q)^{z+p^j}v_Q(P)^{p^{2j}}}\right)^{(p^k-1)/r}. \end{aligned} \tag{23}$$

3.4.1 The super-optimal pairing on the curves with the lack of twists

We first look at examples of pairing-friendly curves E_1, E_2 with the lack of twists. Then the subgroup $\mathbb{G}_2$ can only be represented as $\mathbb{G}_2 = E(\mathbb{F}_{p^k})[r]\cap\ker(\pi-[p])$.

Consequently, the techniques of twist and denominator elimination can not be utilized. In other words, all the vertical line functions can not vanish in the final exponentiation, and more operations need to be performed in the whole extension field $\mathbb{F}_{p^k}$.

Based on the above analysis, the embedding degree k must satisfies $\gcd(k, \#Aut(E)) = 1$. Recall that the curve E_1 (resp. E_2) corresponds to a pairing-friendly curve in Cyclo (6.6) (resp. Cyclo (6.2)) [15]. Recall that for $Q \in \mathbb{G}_2 \subseteq E_i(\mathbb{F}_{p^k})$ there is a positive integer j $(1 \leq j < k)$ such that $\pi^j \circ \sigma(Q) = [z]Q$. If $\mathrm{ord}(\sigma) \nmid k$, it is equipped with the super-optimal ate pairing. The corresponding formula in level 2 is presented in Theorem 3.

Theorem 3. *Using the notation as above, let $(p(z), r(z), t(z))$ represent a family of GLV-curves with the lack of twist equipped with embedding degree k and efficiently-computable automorphism σ such that $ord(\sigma) \nmid k$. Let E_i $(i = 1, 2)$ be a curve in this family, $Q \in E_i(\mathbb{F}_{p^k})[r] \cap \ker(\pi - [p])$, $P \in E_i(\mathbb{F}_p)[r]$. Then the super-optimal ate pairing on $K_i = E_i/\langle \pm 1 \rangle$ $(i = 1, 2)$ corresponding to biextension $X_{2(\mathcal{O}_E)}$ with sections (X, Z) can be executed as follows.*

1. *If we are on $K_1 = E_1/\langle \pm 1 \rangle$ with $j(E_1) = 0$, then the formula for the super-optimal ate pairing is*

$$sopt_b(P, Q) = \left(\frac{Z^z_{[z]Q+P} \cdot Z_{\tau([z]Q)+\pi^j([z]Q)+P}}{Z^z_{[z]Q} \cdot v_Q(P)^{2p^{2j}}} \right)^{(p^k-1)/r}.$$

2. *If we are on $K_2 = E_2/\langle \pm 1 \rangle$ with $j(E_2) = 1728$, then the formula for the super-optimal ate pairing is*

$$sopt_b(P, Q) = \left(\frac{Z^z_{[z]Q+P} \cdot Z^{p^j}_{[z]Q+\sigma^{-1}(P)}}{Z^{z+p^j}_{[z]Q} \cdot v_Q(P)^{2p^{2j}}} \right)^{\frac{p^k-1}{r}}.$$

Proof. For K_1, this follows from Eq. (22), replacing the level 1 cubical arithmetic by level 2 cubical arithmetic. The proof is similar for K_2 from Eq. (23). □

According to the above analysis, we mainly need to compute the coordinates $Z_{[z]Q}$, $Z_{[z]Q+P}$ and $Z_{[z]Q+\sigma^{-1}(P)}$. This can also done by the cubical or double-and-add ladder. Now we present the following family BW13 for description.

Example 5 (BW13 family)**.** *From [10], the BW13 family allows computing super-optimal ate pairing. Besides, it is relevant for the ETNFS attack. Consequently, this family is also an alternative consideration in pairing-based cryptography. If the CM discriminant $D = 1$, the parametrized polynomials $r(z)$, $t(z)$ and $p(z)$ of BW13 are (see Cyclo (6.2) in [15] for more details)*

$$\begin{aligned} r(z) &= \Phi_{52}(z), \\ t(z) &= -z^2 + 1, \\ p(x) &= \frac{1}{4}(z^{30} + 2z^{28} + z^{26} + z^4 - 2z^2 + 1). \end{aligned}$$

Additionally, it is satisfied that $z^2+p \equiv 0 \mod r$ *and* $\pi^7 \circ \sigma(Q) = [z]Q$ *for* $Q \in \mathbb{G}_2$. *By Theorem 3 the super-optimal ate pairing on the above family can be obtained through the cubical arithmetic as*

$$sopt_b(P,Q) = \left(\frac{Z^z_{[z]Q+P} \cdot Z^{p^7}_{[z]Q+\sigma^{-1}(P)}}{Z^{z+p^7}_{[z]Q} \cdot v_Q(P)^{2p^7}} \right)^{\frac{p^{13}-1}{r}}.$$

As for $D = 3$, *the following three polynomials parameterize a family of pairing-friendly curves with embedding degree* $k = 13$ (*see Cyclo* (6.6) *[15] for more details*)

$$\begin{aligned} r(z) &= \Phi_{78}(z), \\ t(z) &= -z^{14} + z + 1, \\ p(x) &= \frac{1}{3}(z+1)^2(z^{26} - z^{13} + 1) - z^{27}. \end{aligned}$$

It can be deduced that for $Q \in \mathbb{G}_2$, *we have* $z^2+zp+p^2 \equiv 0 \mod r$ *and* $\pi \circ \sigma(Q) = [z]Q$. *By Theorem 3, the formula of the super-optimal on this family is*

$$sopt_b(P,Q) = \left(\frac{Z^z_{[z]Q+P} \cdot Z_{\tau([z]Q)+\pi([z]Q)+P}}{Z^z_{[z]Q} \cdot v_Q(P)^{2p^2}} \right)^{\frac{p^{13}-1}{r}}.$$

3.4.2 The super-optimal ate pairing on curves admitting twists

When the embedding degree is even, we first remark that we can simplify the formulas in the previous section. We only treat the case of E_1, the proofs are the same for E_2.

Lemma 9. *When the embedding degree is even, starting with normalized biextension functions* $g_{Q,\sigma^{-1}(P)}, g_{Q,P}$, *we have*

$$sopt(P,Q) = \left(g^{p^j}_{[z]Q,\sigma^{-1}(P)}(\mathcal{O}_E) \cdot g^{p^j+z}_{[z]Q,P}(\mathcal{O}_E) \cdot g_{[p^{2j}]Q,[p^j z]Q}(P) \right)^{(p^k-1)/r}. \tag{24}$$

Plugging the cubical formulas in Eq. (24), *we obtain*

$$sopt(P,Q) = \left(\frac{Z_1([z]Q+P)^z}{Z_1([z]Q)^z} \cdot Z_1(\tau([z]Q) + \pi^j([z]Q) + P) \right)^{(p^k-1)/r}.$$

Proof. This is the formulas from Lemmas 7 and 8, using that when the embedding degree is even, $v_{\pi^{2j}(Q)}(P)$ will be killed by the final exponentiation. □

If $\phi : E' \to E$ is the degree-d twisting isomorphism, we denote $\sigma' = \phi^{-1} \circ \sigma \circ \phi$, and $\tau' = \phi^{-1} \circ \tau \circ \phi = \pi'^j \circ \sigma'$. Beware of the notation, here $\pi' = \phi^{-1} \circ \pi \circ \phi$ is the pullback of the Frobenius on E', it differs from the Frobenius π' on E' by the isomorphism σ' inducing the twist: $\pi = \phi^{-1} \circ \sigma' \circ \pi' \circ \phi$ (see the definitions in Section 3.1). As usual, we

let $P' = \phi^{-1}(P), Q' = \phi^{-1}(Q)$. With these notations, using that ϕ is an isomorphism over $\mathbb{F}_{p^k}$, we immediately have:

Lemma 10. *For any $\mathbb{F}_{p^k}$-rational biextension function $g_{Q',P'}$, we have:*

$$sopt(P,Q) = \left(\tau' \cdot g_{[z]Q',\sigma'^{-1}(P')} \star_1 \pi^j \cdot g_{[z]Q',P'} \star_1 \pi^{2j} \cdot g_{Q',P'}\right)^{\frac{p^k-1}{r}},$$

where $g_{Q',\sigma'^{-1}(P)} = \tau'^{-1} \cdot g_{[z]Q',P'}$. Starting with normalized biextension functions $g_{Q',\sigma'^{-1}(P')}, g_{Q',P'}$, and $g_{[p^{2j}]Q',[p^j z]Q'}$, we have:

$$sopt(P,Q) = \left(g^{p^j}_{[z]Q',\sigma'^{-1}(P')}(\mathcal{O}_E) g^{p^j+z}_{[z]Q',P'}(\mathcal{O}_E) g_{[p^{2j}]Q',[p^j z]Q'}(P)\right)^{(p^k-1)/r}.$$

Moving to level 2 cubical arithmetic, we obtain the following theorem.

Theorem 4. *Using the notation above, let $(p(z), r(z), t(z))$ represent a family of GLV-curves with embedding degree k and an efficiently-computable automorphism σ such that $ord(\sigma) \nmid k$. Let E_i $(i = 1, 2)$ be a curve in this family with a degree-d twist E'_i such that there exists a positive integer j $(1 \leq j < k)$ satisfying $\tau(Q) = [z]Q$ for any $Q \in \mathbb{G}_2$. Assume that $P \in E_i(\mathbb{F}_p)[r]$. Let P' and Q' denote $\phi^{-1}(P)$ and $\phi^{-1}(Q)$, respectively. Then, the super-optimal pairing on $K_i = E_i/\langle \pm 1 \rangle$ $(i = 1, 2)$ corresponding to the biextension $X_{2(\mathcal{O}_E)}$ with sections (X, Z) can be executed as follows*

1. *If we are on $K_1 = E_1/\langle \pm 1 \rangle$ with $j(E_1) = 0$, then the formula for the super-optimal pairing is*

$$sopt_b(P,Q) = \left(Z^z_{[z]Q'+P'} \cdot Z_{\tau'([z]Q')+\pi^j([z]Q')+P'}\right)^{(p^k-1)/r}.$$

2. *If we are on $K_2 = E_2/\langle \pm 1 \rangle$ with $j(E_2) = 1728$, then the formula for the super-optimal pairing is*

$$sopt_b(P,Q) = sopt(P,Q)^2 = \left(Z^z_{[z]Q'+P'} \cdot Z^{p^j}_{[z]Q'+\sigma'^{-1}(P')}\right)^{\frac{p^k-1}{r}}.$$

Proof. For K_1, we could invoke Lemma 10 directly, and redo the same computations that we did for Lemma 8. The difference is that the denominator $Z^{\bullet}_{[z]Q'}$ vanishes in the final exponentiation. The proof is similar for K_2. □

Example 6 (BW14 family)**.** *As mentioned in [9], the BW family with embedding degree $k = 14$ allows computing the pairing in $\log_2(r)/2\varphi(k)$ basic Miller iterations, which makes it a strong candidate in pairing-based cryptography. If CM-discriminant $D = 1$, the corresponding parametrized polynomials $r(z), t(z)$ and $p(z)$ are stated as*

$$r(z) = \Phi_{28}(z),\ t(z) = z^2 + 1,$$
$$p(x) = \frac{1}{4}(z^{18} - 2z^{16} + z^{14} + z^4 + 2z^2 + 1).$$

There exists a quadratic twist E_2' for a BW14 curve E_2. From Section 2.3, the twisting isomorphism is defined as $\phi : E_2' \to E_2,\ (x,y) \mapsto (Dx, D^{\frac{3}{2}}y)$ with $D \in \mathbb{F}_{p^7}^*$ mod $(\mathbb{F}_{p^7}^*)^2$. Additionally, it is satisfied that $z^2 - p \equiv 0 \mod r$ and $\pi^4 \circ \sigma(Q) = [z]Q$ for $Q \in \mathbb{G}_2$ on BW14 family. By Theorem 4 the super-optimal ate pairing on BW14 can be obtained through biextension as

$$sopt_b(P,Q) = sopt(P,Q)^2 = \left(Z^z_{[z]Q'+P'} \cdot Z^{p^4}_{[z]Q'+\sigma^{-1}(P')}\right)^{\frac{p^{14}-1}{r}}.$$

For the curves E_1 with $D = 3$, the family of pairing-friendly curves with embedding degree $k = 14$ (see Cyclo (6.6) [15] for more details) is presented as follows

$$\begin{aligned} r(z) &= \Phi_{42}(z),\ t(z) = -z^8 + z + 1, \\ p(x) &= \frac{1}{3}(z-1)^2(z^{14} - z^7 + 1) + z^{15}. \end{aligned}$$

We deduce that for $Q \in \mathbb{G}_2$, we have $z^2 + zp + p^2 \equiv 0 \mod r$ and $\pi \circ \sigma(Q) = [z]Q$. From Theorem 4, the formula for the super-optimal pairing on this family is

$$sopt_b(P,Q) = \left(Z^z_{[z]Q'+P'} \cdot Z_{\tau'([z]Q')+\pi([z]Q')+P'}\right)^{(p^{14}-1)/r}.$$

4 Computational procedure and cost analysis

In this subsection, we provide the details for the implementation of pairing computations through biextension on different families including BLS12, AFG16, BW14 and BW13. A concrete cost analysis is also presented. Moreover, we compare the corresponding computational costs by employing our algorithms to the approaches in [30] and the popular Miller's algorithm. According to the analysis in Section 3, we present the formulas of the pairing computation by utilizing biextension, for some well-known families of pairing-friendly curves in Table 1.

Table 1 The pairing formulas by exploiting biextension with respect to divisor $2(\mathcal{O}_E)$. The scalar z, the maps ϕ and σ are the parametrized seed, the twisting isomorphism together with the efficiently-computable automorphism of the family of pairing-friendly curves, respectively. Denote by P' and Q' the image points $\phi^{-1}(P)$ and $\phi^{-1}(Q)$, respectively.

k	Curve	Pairing formula via biextension
12	BLS12, $D=3$	$Z_{[z]Q'+P'}^{\frac{p^{12}-1}{r}}$
16	AFG16, $D=1$	$Z_{[z]Q'+P'}^{\frac{p^{16}-1}{r}}$
14	BW14, $D=1$	$\left(Z_{[z]Q'+P'}^{z} \cdot Z_{[z]Q'+\sigma^{-1}(P')}^{p^4}\right)^{\frac{p^{14}-1}{r}}$
14	BW14, $D=3$	$\left(Z_{[z]Q'+P'}^{z} \cdot Z_{\tau'([z]Q')+\pi([z]Q')+P'}\right)^{(p^{14}-1)/r}$
13	BW13, $D=1$	$\left(\frac{Z_{[z]Q+P}^{z} \cdot Z_{[z]Q+\sigma^{-1}(P)}^{p^7}}{Z_{[z]Q}^{z+p^7} \cdot v_Q(P)^{2p^7}}\right)^{\frac{p^{13}-1}{r}}$
13	BW13, $D=3$	$\left(\frac{Z_{[z]Q+P}^{z} \cdot Z_{\tau([z]Q)+\pi([z]Q)+P}}{Z_{[z]Q}^{z} \cdot v_Q(P)^{2p^2}}\right)^{\frac{p^{13}-1}{r}}$

In the following, we provide the detailed computational procedure and cost analysis for the formulas above. We first present the corresponding notations.

Notations. Let $\mathbf{m}$, $\mathbf{s}$, and $\mathbf{i}$ denote the costs of multiplication, squaring and inversion in $\mathbb{F}_p$, respectively. Let $\mathbf{m}_k$, $\mathbf{s}_k$, $\mathbf{i}_k$ and $\mathbf{f}_k$ represent the costs of addition, multiplication, squaring, inversion and Frobenius endomorphism in $\mathbb{F}_{p^k}$, respectively. Denote by $\mathbf{m}_0$ the cost of multiplication by a constant. We omit the calculations of the additions and subtractions over finite fields for simplicity.

4.1 Computational procedure and cost analysis for ate pairing on BLS12 family

In this subsection, we focus on the computation process and cost calculation for the ate pairing via biextension on BLS12 family. The extension field $\mathbb{F}_{p^{12}}$ can be constructed as follows

$$\mathbb{F}_p \Rightarrow \mathbb{F}_{p^2} = \mathbb{F}_p[u]/(u^2-\alpha) \Rightarrow \mathbb{F}_{p^6} = \mathbb{F}_{p^2}[v]/(v^3-u) \Rightarrow \mathbb{F}_{p^{12}} = \mathbb{F}_{p^6}[w]/(w^2-v),$$

where $\alpha \in \mathbb{F}_p$. From Table 1 we can see that it needs to compute

$$a_{b,\lambda}(P,Q) = Z_{[z]Q'+P'}^{\frac{p^{12}-1}{r}},$$

where ϕ is a degree-6 twist isomorphism

$$\phi : E' \to E,\ (x, y) \mapsto (xv, yvw).$$

The equations of E and E' are $y^2 = x^3 + b$ and $y^2 = x^3 + b/u$, respectively. As mentioned in Section 2.4, we can employ the cubical ladder or double-and-add ladder to derive $Z_{[z]Q'+P'}$.

Denote by $\texttt{cDBL}(P)$, $\texttt{cDIFF}(P, Q, iX_{P-Q})$ and $\texttt{cADD}(P_1, P_2, P_1 + Q, P_2 - Q)$ the x-only cubical point doubling, differential addition and compatible addition (See Algorithms 7, 8 and 11 for more details) on the Kummer line $K = E/\langle \pm 1 \rangle$ with $j(E) = 0$, respectively. The detailed computational procedure for the cubical ladder is presented in Algorithm 1.

Algorithm 1 The cubical ladder to compute $Z_{[z]Q'+P'}$

Input: The points $Q' = (X_{Q'} : Z_{Q'})$, $P' = (X_{P'} : Z_{P'})$, $Q' + P' = (X_{Q'+P'} : Z_{Q'+P'}) \in E'$. The inverses of the X-coordinates of Q', P' and $Q' - P'$: $iX_{Q'}$, $iX_{P'}$, $iX_{Q'-P'}$. The scalar $z = \sum_{i=0}^{N} n_i 2^i$ $(n > 2)$.
Output: The point $[z]Q' + P' = (X_{[z]Q'+P'} : Z_{[z]Q'+P'})$

1: $R \leftarrow Q'$, $S \leftarrow \texttt{cDBL}(Q')$, $T \leftarrow \texttt{cDIFF}(P' + Q', Q', iX_{P'})$
2: **for** $i = N - 1$ **to** 0 **do** ▷ $R = [n]Q'$, $S = [n+1]Q'$, $T = [n]Q' + P'$
3: $\quad U \leftarrow \texttt{cDIFF}(S, R, iX_{Q'})$
4: $\quad$ **if** $n_i = 0$ **then**
5: $\quad\quad T \leftarrow \texttt{cDIFF}(T, R, iX_{P'})$
6: $\quad\quad R \leftarrow \texttt{cDBL}(R)$
7: $\quad\quad S \leftarrow U$
8: $\quad$ **else**
9: $\quad\quad T \leftarrow \texttt{cDIFF}(S, T, iX_{Q'-P'})$
10: $\quad\quad S \leftarrow \texttt{cDBL}(S)$
11: $\quad\quad R \leftarrow U$
12: $\quad$ **end if**
13: **end for**
14: **return** T

According to Algorithm 1, it requires to execute a cubical point doubling and two differential additions per step. More precisely, the point doubling (Lines 6 and 10 in Algorithm 1) and one of the differential additions (Line 3 in Algorithm 1) are performed in $E'(\mathbb{F}_{p^2})$, while the other differential addition (Lines 5 and 9 in Algorithm 1) is accomplished over $\mathbb{F}_{p^{12}}$.

Note that in the phase of the $\texttt{cDBL}$ over $\mathbb{F}_{p^2}$, the coefficient of E' is $b' = b/u$. If b is small, multiplying an element by b' can be regarded as a shifting operation, whose cost is negligible. According to Algorithms 7 and 8, the corresponding costs for the cubical point doubling and differential addition over $\mathbb{F}_{p^2}$ are respectively

$$\text{Cost}_{\texttt{cDBL}} = 4\mathbf{m}_2 + 2\mathbf{s}_2 \text{ and } \text{Cost}_{\texttt{cDIFF}_{\mathbb{F}_{p^2}}} = 6\mathbf{m}_2 + 2\mathbf{s}_2.$$

Additionally, if the bit is 0, the multiplication by $1/X_{P'}$ in cDIFF can also be regarded as a shifting operation over $\mathbb{F}_{p^{12}}$. Besides, the cost of the operation for multiplying two elements in $\mathbb{F}_{p^2}$ and $\mathbb{F}_{p^{12}}$ can be estimated as $6\mathbf{m}_2$. From Algorithm 8, the costs for Lines 5 and 9 in Algorithm 1 are respectively

$$\mathrm{Cost}_{\mathtt{cDIFF}_{\mathrm{bit0}}} = \mathbf{m}_{12} + 2\mathbf{s}_{12} + 4 \cdot 6\mathbf{m}_2 \text{ and } \mathrm{Cost}_{\mathtt{cDIFF}_{\mathrm{bit1}}} = 2\mathbf{m}_{12} + 2\mathbf{s}_{12} + 4 \cdot 6\mathbf{m}_2.$$

On this basis, the computational cost for an iteration of the cubical ladder is

$$\begin{aligned}\mathrm{Cost}_{\mathrm{cubic0}} &= \mathrm{Cost}_{\mathtt{cDBL}} + \mathrm{Cost}_{\mathtt{cDIFF}_{\mathbb{F}_{p^2}}} + \mathrm{Cost}_{\mathtt{cDIFF}_{\mathrm{bit0}}} = \mathbf{m}_{12} + 2\mathbf{s}_{12} + 34\mathbf{m}_2 + 4\mathbf{s}_2,\\ \mathrm{Cost}_{\mathrm{cubic1}} &= \mathrm{Cost}_{\mathtt{cDBL}} + \mathrm{Cost}_{\mathtt{cDIFF}_{\mathbb{F}_{p^2}}} + \mathrm{Cost}_{\mathtt{cDIFF}_{\mathrm{bit1}}} = 2\mathbf{m}_{12} + 2\mathbf{s}_{12} + 34\mathbf{m}_2 + 4\mathbf{s}_2.\end{aligned}$$

The concrete computational process of the double-and-add ladder is illustrated in Algorithm 2.

Algorithm 2 The double-and-add ladder to compute $Z_{[z]Q'+P'}$

Input: The points $Q' = (X_{Q'} : Z_{Q'})$, $P' = (X_{P'} : Z_{P'})$, $Q' + P' = (X_{Q'+P'} : Z_{Q'+P'}) \in E'$. The inverses of the X-coordinates of Q', P' and $Q' - P'$: $iX_{Q'}$, $iX_{P'}$, $iX_{Q'-P'}$. The scalar $z = \sum_{i=0}^{N} n_i 2^i$.
Output: The point $[z]Q' + P' = (X_{[z]Q'+P'} : Z_{[z]Q'+P'})$

1: $R \leftarrow Q'$, $S \leftarrow Q' + P'$
2: **for** $i = N - 1$ **to** 0 **do** ▷ $R = [n]Q'$, $S = [n]Q' + P'$
3: **if** $n_i = 0$ **then**
4: $R \leftarrow \mathtt{cDBL}(R)$
5: $S \leftarrow \mathtt{cDIFF}(S, R, iX_{P'})$
6: **else**
7: $T \leftarrow \mathtt{cADD}(R, Q', S, Q' - P')$ ▷ $T = [n+1]Q'$
8: $R \leftarrow \mathtt{cDIFF}(T, R, iX_{Q'})$
9: $S \leftarrow \mathtt{cDIFF}(T, S, iX_{Q'-P'})$
10: **end if**
11: **end for**
12: **return** S

Now we analyze the cost for each basic iteration step in Algorithm 2. Based on the above analysis, the cost for a doubling step is

$$\mathrm{Cost}_{\mathrm{dbl}} = \mathrm{Cost}_{\mathtt{cDBL}} + \mathrm{Cost}_{\mathtt{cDIFF}_{\mathrm{bit0}}} = \mathbf{m}_{12} + 2\mathbf{s}_{12} + 28\mathbf{m}_2 + 2\mathbf{s}_2.$$

As for a double-and-add step, it requires one compatible addition, together with two differential additions to perform. One of the differential additions (Line 8 in Algorithm 2) is executed over $\mathbb{F}_{p^2}$, while the other (Line 9 in Algorithm 2) is over $\mathbb{F}_{p^{12}}$. It is worth noting that part of the operations during the compatible addition are carried out over

$\mathbb{F}_{p^{12}}$. From Algorithm 11 we can calculate the cost for the compatible addition as follows

$$\text{Cost}_{\texttt{cADD}} = 4\mathbf{m}_{12} + 3\mathbf{s}_{12} + 28\mathbf{m}_2 + 3\mathbf{s}_2.$$

Therefore, the computational cost for a double-and-add iteration step is

$$\text{Cost}_{\text{dbladd}} = \text{Cost}_{\texttt{cDIFF}_{\mathbb{F}_{p^2}}} + \text{Cost}_{\texttt{cDIFF}_{\text{bit1}}} + \text{Cost}_{\texttt{cADD}} = 6\mathbf{m}_{12} + 5\mathbf{s}_{12} + 58\mathbf{m}_2 + 5\mathbf{s}_2.$$

4.2 Computational procedure and cost analysis for optimal ate pairing on AFG16 family

In this subsection, we explore to derive the concrete computational procedure and cost analysis for the optimal ate pairing on family AFG16 with $D = 1$. The field $\mathbb{F}_{p^{16}}$ can be constructed as

$$\mathbb{F}_p \Rightarrow \mathbb{F}_{p^4} = \mathbb{F}_p[u]/(u^4 - \alpha) \Rightarrow \mathbb{F}_{p^8} = \mathbb{F}_{p^4}[v]/(v^2 - u) \Rightarrow \mathbb{F}_{p^{16}} = \mathbb{F}_{p^8}[w]/(w^2 - v),$$

where $\alpha \in \mathbb{F}_p$. The degree-4 twist isomorphism ϕ on this family is defined as follows

$$\phi : E' \to E,\ (x, y) \mapsto (xv, yvw),$$

where the curve E and its twist E' are given by $y^2 = x^3 + ax$ and $y^2 = x^3 + a/u \cdot x$, respectively. Recalled from Table 1 and Example 4, the optimal pairing on AFG16 family via biextension can be derived as

$$opt_b(P, Q) = Z_{[z]Q'+P'}^{\frac{p^{16}-1}{r}}.$$

Hence, it requires to compute $Z_{[z]Q'+P'}$, which can also be done by exploiting Algorithm 1 or 2. For `cDBL` and `cADD` performed over $\mathbb{F}_{p^4}$, the coefficient of E' is $a' = a/u$. Thus the cost of multiplying an element by a' can be omitted if a is small. From Algorithms 11 and 12, the costs for `cDBL` and `cADD` on Kummer line $K = E'/\langle \pm 1 \rangle$ with $j(E') = 1728$ over $\mathbb{F}_{p^4}$ are respectively

$$\text{Cost}_{\texttt{cDBL}} = 2\mathbf{m}_4 + 3\mathbf{s}_4 \text{ and } \text{Cost}_{\texttt{cDIFF}_{\mathbb{F}_{p^4}}} = 4\mathbf{m}_4 + 2\mathbf{s}_4.$$

Moreover, the cost for multiplying two elements in $\mathbb{F}_{p^4}$ and $\mathbb{F}_{p^{16}}$ can be estimated as $4\mathbf{m}_4$. Consequently, from Algorithm 12 the costs for the cubical differential additions (Lines 5 and 9 in Algorithm 1) are

$$\text{Cost}_{\texttt{cDIFF}_{\text{bit0}}} = 2\mathbf{s}_{16} + 3 \cdot 4\mathbf{m}_4,\ \text{Cost}_{\texttt{cDIFF}_{\text{bit1}}} = \mathbf{m}_{16} + 2\mathbf{s}_{16} + 3 \cdot 4\mathbf{m}_4.$$

Based on the above analysis, the computational cost for an iteration in Algorithm 1 is

$$\text{Cost}_{\text{cubic0}} = \text{Cost}_{\texttt{cDBL}} + \text{Cost}_{\texttt{cDIFF}_{\mathbb{F}_{p^4}}} + \text{Cost}_{\texttt{cDIFF}_{\text{bit0}}} = 2\mathbf{s}_{16} + 18\mathbf{m}_4 + 5\mathbf{s}_4.$$
$$\text{Cost}_{\text{cubic1}} = \text{Cost}_{\texttt{cDBL}} + \text{Cost}_{\texttt{cDIFF}_{\mathbb{F}_{p^4}}} + \text{Cost}_{\texttt{cDIFF}_{\text{bit1}}} = \mathbf{m}_{16} + 2\mathbf{s}_{16} + 18\mathbf{m}_4 + 5\mathbf{s}_4.$$

We now present the cost calculation for the double-and-add ladder (Algorithm 2). From the previous analysis, the cost for a doubling step is

$$\mathrm{Cost}_{\mathrm{dbl}} = \mathrm{Cost}_{\mathtt{cDBL}} + \mathrm{Cost}_{\mathtt{cDIFF}_{\mathrm{bit0}}} = 2\mathbf{s}_{16} + 14\mathbf{m}_4 + 3\mathbf{s}_4.$$

As for a double-and-add step, we need to execute two cubical differential additions (over $\mathbb{F}_{p^4}$ and $\mathbb{F}_{p^{16}}$) along with a compatible addition. According to Algorithms 10 and 12, the corresponding computational costs are

$$\begin{aligned}&\mathrm{Cost}_{\mathtt{cDIFF}_{\mathbb{F}_{p^4}}} = 4\mathbf{m}_4 + 2\mathbf{s}_4,\ \mathrm{Cost}_{\mathtt{cDIFF}_{\mathrm{bit1}}} = \mathbf{m}_{16} + 2\mathbf{s}_{16} + 3\cdot 4\mathbf{m}_4,\\ &\mathrm{Cost}_{\mathtt{cADD}} = 4\mathbf{m}_{16} + \mathbf{s}_{16} + 19\mathbf{m}_4 + \mathbf{s}_4.\end{aligned}$$

On this basis, the computational cost for a double-and-add step in Algorithm 2 is

$$\mathrm{Cost}_{\mathrm{dbladd}} = \mathrm{Cost}_{\mathtt{cDIFF}_{\mathbb{F}_{p^4}}} + \mathrm{Cost}_{\mathtt{cDIFF}_{\mathrm{bit1}}} + \mathrm{Cost}_{\mathtt{cADD}} = 5\mathbf{m}_{16} + 3\mathbf{s}_{16} + 35\mathbf{m}_4 + 3\mathbf{s}_4.$$

4.3 Implementation detail and cost analysis for super-optimal ate pairing on BW family

We now investigate the concrete computational processes for the super-optimal ate pairings on families BW14 and BW13 via biextension. Besides, we also present the computational cost analysis for each iteration of the biextension ladders. For simplicity, we only provide the technical details for the pairing-friendly curves with CM-discriminant $D = 1$.

4.3.1 Super-optimal ate pairings on BW14 family

In this subsection, we first explore to derive the algorithm for the super-optimal ate pairing on family BW14 with $D = 1$ utilizing biextension. The field $\mathbb{F}_{p^{14}}$ can be constructed as

$$\mathbb{F}_p \Rightarrow \mathbb{F}_{p^7} = \mathbb{F}_p[u]/(u^7 - \alpha) \Rightarrow \mathbb{F}_{p^{14}} = \mathbb{F}_{p^7}[v]/(v^2 - u),$$

where $\alpha \in \mathbb{F}_p$. According to Table 1, the super-optimal ate pairing on BW14 family with $D = 1$ through biextension can be derived as

$$sopt_b(P, Q) = \left(Z^z_{[z]Q'+P'} \cdot Z^{p^4}_{[z]Q'+\sigma^{-1}(P')}\right)^{\frac{p^{14}-1}{r}},$$

where ϕ is a degree-2 twist isomorphism

$$\phi : E' \to E,\ (x, y) \mapsto (xu, yuv).$$

The curve E and its twist E' are defined as $y^2 = x^3 + ax$ and $y^2 = x^3 + a/u \cdot x$, respectively. Moreover, as mentioned in Section 2.1, E is equipped with an efficiently-computable automorphism $\sigma : (x, y) \mapsto (-x, \beta y)$, where $\beta \in \mathbb{F}_p$ satisfies $\beta^2 =$

-1. From the above formula, we need to compute two coordinates $Z_{[z]Q'+P'}$ and $Z_{[z]Q'+\sigma^{-1}(P')}$. An intuitive approach is to separately calculate them by Algorithm 1 (or Algorithm 2). Nevertheless, it is worth noting that part of the computation can be shared. In the following, we describe how to share the information during the computations of $Z_{[z]Q'+P'}$ and $Z_{[z]Q'+\sigma^{-1}(P')}$.

In each iteration step of the cubical (resp. double-and-add) ladder in Algorithm 1 (resp. Algorithm 2), the $(X : Z)$-coordinates of $[k]Q'$ are both needed in the phase of computing $Z_{[z]Q'+P'}$ and $Z_{[z]Q'+\sigma^{-1}(P')}$. Consequently, we can accomplish the computation of these two coordinates in the same ladder. The shared cubical and double-and-add ladders are presented in Algorithms 3 and 4, respectively.

Algorithm 3 The shared cubical ladder

Input: The points $Q' = (X_{Q'} : Z_{Q'})$, $P' = (X_{P'} : Z_{P'})$, $Q' + P' = (X_{Q'+P'} : Z_{Q'+P'})$ and $Q' + \sigma^{-1}(P') = (X_{Q'+\sigma^{-1}(P')} : Z_{Q'+\sigma^{-1}(P')}) \in E'$. The inverses of the X-coordinates of Q', P', $Q' - P'$ and $Q' - \sigma^{-1}(P')$: $iX_{Q'}$, $iX_{P'}$, $iX_{Q'-P'}$ and $iX_{Q'-\sigma^{-1}(P')}$. The scalar $z = \sum_{i=0}^{N} n_i 2^i$.

Output: The points $[z]Q' + P' = (X_{[z]Q'+P'} : Z_{[z]Q'+P'})$ and $[z]Q' + \sigma^{-1}(P') = (X_{[z]Q'+\sigma^{-1}(P')} : Z_{[z]Q'+\sigma^{-1}(P')})$.

1: $R \leftarrow Q'$, $S \leftarrow \texttt{cDBL}(Q')$, $T_1 \leftarrow \texttt{cDIFF}(P' + Q', Q', iX_{P'})$
2: $T_2 \leftarrow \texttt{cDIFF}(\sigma^{-1}(P') + Q', Q', -iX_{P'})$
3: **for** $i = N-1$ **to** 0 **do** ▷ $R = [k]Q', S = [k+1]Q', T_1 = [k]Q'+P', T_2 = [k]Q'+\sigma^{-1}(P')$
4: $\quad U \leftarrow \texttt{cDIFF}(S, R, iX_{Q'})$
5: $\quad$ **if** $n_i = 0$ **then**
6: $\quad\quad T_1 \leftarrow \texttt{cDIFF}(T_1, R, iX_{P'})$
7: $\quad\quad T_2 \leftarrow \texttt{cDIFF}(T_2, R, -iX_{P'})$
8: $\quad\quad R \leftarrow \texttt{cDBL}(R)$
9: $\quad\quad S \leftarrow U$
10: $\quad$ **else**
11: $\quad\quad T_1 \leftarrow \texttt{cDIFF}(S, T_1, iX_{Q'-P'})$
12: $\quad\quad T_2 \leftarrow \texttt{cDIFF}(S, T_2, iX_{Q'-\sigma^{-1}(P')})$
13: $\quad\quad S \leftarrow \texttt{cDBL}(S)$
14: $\quad\quad R \leftarrow U$
15: $\quad$ **end if**
16: **end for**
17: **return** T_1, T_2

We now make a cost analysis for each iteration step in these two ladders. During an iteration, the operation of multiplying an element by $1/X_{P'}$ over $\mathbb{F}_{p^{14}}$ can be regarded as a shifting in cDIFF if the bit is 0 (Lines 6 and 7 in Algorithm 3, or Lines 5 and 6 in Algorithm 4). And the cost for multiplying two elements in $\mathbb{F}_{p^7}$ and $\mathbb{F}_{p^{14}}$ can be estimated as $2\mathbf{m}_7$. Consequently, according to Algorithm 10, the computational costs for the cubical differential additions (Lines 6, 7, 11, 12 in Algorithm 3, or Lines 5, 6,

Algorithm 4 The shared double-and-add ladder

Input: The points $Q' = (X_{Q'} : Z_{Q'})$, $P' = (X_{P'} : Z_{P'})$, $Q' + P' = (X_{Q'+P'} : Z_{Q'+P'})$ and $Q' + \sigma^{-1}(P') = (X_{Q'+\sigma^{-1}(P')} : Z_{Q'+\sigma^{-1}(P')}) \in E'$. The inverses of the X-coordinates of $Q', P', Q' - P'$ and $Q' - \sigma^{-1}(P')$: $iX_{Q'}$, $iX_{P'}$, $iX_{Q'-P'}$ and $iX_{Q'-\sigma^{-1}(P')}$. The scalar $z = \sum_{i=0}^{N} n_i 2^i$.

Output: The points $[z]Q' + P' = (X_{[z]Q'+P'} : Z_{[z]Q'+P'})$ and $[z]Q' + \sigma^{-1}(P') = (X_{[z]Q'+\sigma^{-1}(P')} : Z_{[z]Q'+\sigma^{-1}(P')})$.

$R \leftarrow Q'$, $S_1 \leftarrow Q' + P'$, $S_2 \leftarrow Q' + \sigma^{-1}(P')$
for $i = N - 1$ **to** 0 **do** ▷ $R = [n]Q', S_1 = [n]Q' + P', S_2 = [n]Q' + \sigma^{-1}(P')$
  if $n_i = 0$ **then**
    $R \leftarrow \texttt{cDBL}(R)$
    $S_1 \leftarrow \texttt{cDIFF}(S_1, R, -iX_{P'})$
    $S_2 \leftarrow \texttt{cDIFF}(S_2, R, -iX_{\sigma^{-1}(P')})$
  else
    $T \leftarrow \texttt{cADD}(R, Q', S_1, Q' - P')$
    $R \leftarrow \texttt{cDIFF}(T, R, iX_{Q'})$
    $S_1 \leftarrow \texttt{cDIFF}(T, S_1, iX_{Q'-P'})$
    $S_2 \leftarrow \texttt{cDIFF}(T, S_2, iX_{Q'-\sigma^{-1}(P')})$
  end if
end for
return S_1, S_2

10, 11 in Algorithm 3) are

$$\text{Cost}_{\texttt{cDIFF}_{\text{bit0}}} = 2\mathbf{s}_{14} + 3 \cdot 2\mathbf{m}_7, \ \text{Cost}_{\texttt{cDIFF}_{\text{bit1}}} = \mathbf{m}_{14} + 2\mathbf{s}_{14} + 3 \cdot 2\mathbf{m}_7.$$

Besides, from Algorithms 9 and 10 the costs for the cubical point doubling and differential addition in $\mathbb{F}_{p^7}$ are

$$\text{Cost}_{\texttt{cDBL}} = 2\mathbf{m}_7 + 3\mathbf{s}_7, \ \text{Cost}_{\texttt{cDIFF}_{\mathbb{F}_{p^7}}} = 4\mathbf{m}_7 + 2\mathbf{s}_7.$$

Therefore, the computational cost for an iteration of the shared cubical ladder (Algorithm 3) is

$$\text{Cost}_{\text{cubic0}} = \text{Cost}_{\texttt{cDBL}} + \text{Cost}_{\texttt{cDIFF}_{\mathbb{F}_{p^7}}} + 2\text{Cost}_{\texttt{cDIFF}_{\text{bit0}}} = 4\mathbf{s}_{14} + 18\mathbf{m}_7 + 5\mathbf{s}_7,$$
$$\text{Cost}_{\text{cubic1}} = \text{Cost}_{\texttt{cDBL}} + \text{Cost}_{\texttt{cDIFF}_{\mathbb{F}_{p^7}}} + 2\text{Cost}_{\texttt{cDIFF}_{\text{bit1}}} = 2\mathbf{m}_{14} + 4\mathbf{s}_{14} + 18\mathbf{m}_7 + 5\mathbf{s}_7.$$

From the previous analysis, the cost for a doubling step in the shared double-and-add ladder (Algorithm 4) is

$$\text{Cost}_{\text{dbl}} = \text{Cost}_{\texttt{cDBL}} + 2\text{Cost}_{\texttt{cDIFF}_{\text{bit0}}} = 4\mathbf{s}_{14} + 14\mathbf{m}_7 + 3\mathbf{s}_7.$$

As for the double-and-add step, we need to execute three differential additions and a compatible addition. By Algorithm 12, the computational cost for the compatible

addition is

$$\mathrm{Cost}_{\mathtt{cADD}} = 4\mathbf{m}_{14} + \mathbf{s}_{14} + 11\mathbf{m}_7 + \mathbf{s}_7.$$

On this basis, the computational cost for a double-and-add step is

$$\mathrm{Cost}_{\mathrm{dbladd}} = \mathrm{Cost}_{\mathtt{cDIFF}_{\mathbb{F}_{p^7}}} + 2\mathrm{Cost}_{\mathtt{cDIFF}_{\mathrm{bit1}}} + \mathrm{Cost}_{\mathtt{cADD}} = 6\mathbf{m}_{14} + 5\mathbf{s}_{14} + 27\mathbf{m}_4 + 3\mathbf{s}_7.$$

4.3.2 The super-optimal ate pairing on BW13 family

In this subsection, we provide the detailed algorithm for the computation of super-optimal pairing on family BW13 with CM-discriminant $D = 1$ [19, Table 5] via biextension. The extension field $\mathbb{F}_{p^{13}}$ can be constructed as

$$\mathbb{F}_p \Rightarrow \mathbb{F}_{p^{13}} = \mathbb{F}_p[u]/(u^{13} - \alpha),$$

where $\alpha \in \mathbb{F}_p$. Different from the pairing-friendly curves discussed before, there exists no twist on BW13 since the embedding degree is a prime. From Table 1 we know that the super-optimal ate pairing on BW13 family with CM-discriminant $D = 1$ through biextension can be obtained as

$$sopt_b(P, Q) = \left(\frac{Z^z_{[z]Q+P} \cdot Z^{p^7}_{[z]Q+\sigma^{-1}(P)}}{Z^{z+p^7}_{[z]Q} \cdot v_Q(P)^{2p^7}} \right)^{\frac{p^{13}-1}{r}},$$

where σ is an efficiently-computable automorphism

$$\sigma : E \to E,\ (x, y) \mapsto (-x, \beta y) \text{ with } \beta^2 + 1 = 0.$$

The cubical and double-and-add ladders can be employed to compute $Z_{[-z]Q+P}$ and $Z_{[-z]Q+\sigma^{-1}(P)}$. The corresponding computational processes are presented in Algorithms 5 and 6.

We now calculate the computational cost for a basic iteration in the cubical and double-and-add ladder. It follows from Algorithms 9 and 10 that the costs of `cDBL` and `cDIFF` on $K = E/\langle \pm 1 \rangle$ over $\mathbb{F}_{p^{13}}$ are

$$\mathrm{Cost}_{\mathtt{cDBL}} = 2\mathbf{m}_{13} + 3\mathbf{s}_{13},\ \mathrm{Cost}_{\mathtt{cDIFF}} = 4\mathbf{m}_{13} + 2\mathbf{s}_{13}.$$

More precisely, some of the cubical differential additions (Lines 6 and 7 in Algorithm 5) involve the operation of multiplying two elements in $\mathbb{F}_p$ and $\mathbb{F}_{p^{13}}$, whose cost can be taken as $13\mathbf{m}$. Consequently, according to Algorithm 10 the cost for this type of cubical differential addition is $\mathrm{Cost}_{\mathtt{cDIFF}_{\mathrm{bit0}}} = 3\mathbf{m}_{13} + 2\mathbf{s}_{13} + 13\mathbf{m}$. On this basis, from Algorithm 5 we know that the computational cost for a basic iteration of the shared cubical ladder is

$$\mathrm{Cost}_{\mathrm{cubic0}} = \mathrm{Cost}_{\mathtt{cDBL}} + \mathrm{Cost}_{\mathtt{cDIFF}_{\mathrm{bit0}}} + 2\mathrm{Cost}_{\mathtt{cDIFF}} = 12\mathbf{m}_{13} + 9\mathbf{s}_{13} + 26\mathbf{m},$$

Algorithm 5 The shared cubical ladder

Input: The points $Q = (X_Q : Z_Q)$, $P = (X_P : Z_P)$, $Q + P = (X_{Q+P} : Z_{Q+P})$ and $Q + \sigma^{-1}(P) = (X_{Q+\sigma^{-1}(P)} : Z_{Q+\sigma^{-1}(P)}) \in E$. The inverses of the X-coordinates of $Q, P, Q - P$ and $Q - \sigma^{-1}(P)$: iX_Q, iX_P, iX_{Q-P} and $iX_{Q-\sigma^{-1}(P)}$. The scalar $z = \sum_{i=0}^{N} n_i 2^i$.

Output: The points $[z]Q + P = (X_{[z]Q+P} : Z_{[z]Q+P})$ and $[z]Q + \sigma^{-1}(P) = (X_{[z]Q+\sigma^{-1}(P)} : Z_{[z]Q+\sigma^{-1}(P)})$.

$R \leftarrow Q$, $S \leftarrow \texttt{cDBL}(Q)$, $T_1 \leftarrow \texttt{cDIFF}(P + Q, Q, iX_P)$
$T_2 \leftarrow \texttt{cDIFF}(\sigma^{-1}(P) + Q, Q, -iX_P)$
for $i = N - 1$ **to** 0 **do** ▷ $R = [k]Q, S = [k + 1]Q, T_1 = [k]Q + P, T_2 = [k]Q + \sigma^{-1}(P)$
    $U \leftarrow \texttt{cDIFF}(S, R, iX_Q)$
    if $n_i = 0$ **then**
        $T_1 \leftarrow \texttt{cDIFF}(T_1, R, iX_P)$
        $T_2 \leftarrow \texttt{cDIFF}(T_2, R, -iX_P)$
        $R \leftarrow \texttt{cDBL}(R)$
        $S \leftarrow U$
    else
        $T_1 \leftarrow \texttt{cDIFF}(S, T_1, iX_{Q-P})$
        $T_2 \leftarrow \texttt{cDIFF}(S, T_2, iX_{Q-\sigma^{-1}(P)})$
        $S \leftarrow \texttt{cDBL}(S)$
        $R \leftarrow U$
    end if
end for
return T_1, T_2

$$\text{Cost}_{\text{cubic1}} = \text{Cost}_{\texttt{cDBL}} + 3\text{Cost}_{\texttt{cDIFF}} = 14\mathbf{m}_{13} + 9\mathbf{s}_{13}.$$

As for the double-and-add step in Algorithm 6, we need to execute one compatible addition, together with three cubical differential additions over $\mathbb{F}_{p^{13}}$. Since the coefficient a is small, from Algorithm 12 the computational cost of `cADD` is about

$$\text{Cost}_{\texttt{cADD}} = 11\mathbf{m}_{13} + 2\mathbf{s}_{13}.$$

On this basis, the computational cost for a basic iteration in the double-and-add ladder (Algorithm 6) is

$$\text{Cost}_{\text{dbl}} = \text{Cost}_{\texttt{cDBL}} + 2\text{Cost}_{\texttt{cDIFF}_{\text{bit0}}} = 8\mathbf{m}_{13} + 7\mathbf{s}_{13} + 26\mathbf{m},$$
$$\text{Cost}_{\text{dbladd}} = \text{Cost}_{\texttt{cADD}} + 3\text{Cost}_{\texttt{cDIFF}} = 23\mathbf{m}_{13} + 8\mathbf{s}_{13}.$$

4.4 Cost comparison

Building upon the analyses presented in Sections 4.1, 4.2 and 4.3, we make a concrete cost comparison for each basic iteration step within the pairing computation between Miller's algorithm and biextension. The cost calculations encompass the Miller iterations on families BLS12, AFG16, BW14 and BW13. Table 2 illustrates the

Algorithm 6 The shared double-and-add ladder

Input: The points $Q = (X_Q : Z_Q)$, $P = (X_P : Z_P)$, $Q + P = (X_{Q+P} : Z_{Q+P})$ and $Q + \sigma^{-1}(P) = (X_{Q+\sigma^{-1}(P)} : Z_{Q+\sigma^{-1}(P)}) \in E$. The inverses of the X-coordinates of Q, P, $Q - P$ and $Q - \sigma^{-1}(P)$: iX_Q, iX_P, iX_{Q-P} and $iX_{Q-\sigma^{-1}(P)}$. The scalar $z = \sum_{i=0}^{N} n_i 2^i$.

Output: The points $[z]Q + P = (X_{[z]Q+P} : Z_{[z]Q+P})$ and $[z]Q + \sigma^{-1}(P) = (X_{[z]Q+\sigma^{-1}(P)} : Z_{[z]Q+\sigma^{-1}(P)})$.

$R \leftarrow Q,\ S_1 \leftarrow Q + P,\ S_2 \leftarrow Q + \sigma^{-1}(P)$
for $i = N - 1$ **to** 0 **do** ▷ $R = [n]Q, S_1 = [n]Q + P, S_2 = [n]Q + \sigma^{-1}(P)$
  if $n_i = 0$ **then**
    $R \leftarrow \texttt{cDBL}(R)$
    $S_1 \leftarrow \texttt{cDIFF}(S_1, R, -iX_P)$
    $S_2 \leftarrow \texttt{cDIFF}(S_2, R, -iX_{\sigma^{-1}(P)})$
  else
    $T \leftarrow \texttt{cADD}(R, Q, S_1, Q - P)$
    $R \leftarrow \texttt{cDIFF}(T, R, iX_Q)$
    $S_1 \leftarrow \texttt{cDIFF}(T, S_1, iX_{Q-P})$
    $S_2 \leftarrow \texttt{cDIFF}(T, S_2, iX_{Q-\sigma^{-1}(P)})$
  end if
end for
return $S_1,\ S_2$

computational costs of each step of the Miller loop using biextension on these families, which are carefully measured and presented, taking into account the properties of each family in the previous subsections.

Table 2 The comparison of the costs of a basic iteration in evaluating the biextension function between utilizing cubical and double-and-add (noted as "dadd") ladders on families BLS12 ($D = 3$), AFG16 ($D = 1$), BW14 ($D = 1$) and BW13 ($D = 1$).

Family	Approach	bit = 0	bit = 1
BLS12	cubical	$\mathbf{m}_{12} + 2\mathbf{s}_{12} + 34\mathbf{m}_2 + 4\mathbf{s}_2$	$2\mathbf{m}_{12} + 2\mathbf{s}_{12} + 34\mathbf{m}_2 + 4\mathbf{s}_2$
	dadd	$\mathbf{m}_{12} + 2\mathbf{s}_{12} + 28\mathbf{m}_2 + 2\mathbf{s}_2$	$6\mathbf{m}_{12} + 5\mathbf{s}_{12} + 58\mathbf{m}_2 + 5\mathbf{s}_2$
AFG16 [19]	cubical	$2\mathbf{s}_{16} + 18\mathbf{m}_4 + 5\mathbf{s}_4$	$\mathbf{m}_{16} + 2\mathbf{s}_{16} + 18\mathbf{m}_4 + 5\mathbf{s}_4$
	dadd	$2\mathbf{s}_{16} + 14\mathbf{m}_4 + 3\mathbf{s}_4$	$5\mathbf{m}_{16} + 3\mathbf{s}_{16} + 35\mathbf{m}_4 + 3\mathbf{s}_4$
BW14 [9]	cubical	$4\mathbf{s}_{14} + 18\mathbf{m}_7 + 5\mathbf{s}_7$	$2\mathbf{m}_{14} + 4\mathbf{s}_{14} + 18\mathbf{m}_7 + 5\mathbf{s}_7$
	dadd	$4\mathbf{s}_{14} + 14\mathbf{m}_7 + 3\mathbf{s}_7$	$6\mathbf{m}_{14} + 5\mathbf{s}_{14} + 27\mathbf{m}_7 + 3\mathbf{s}_7$
BW13 [19]	cubical	$12\mathbf{m}_{13} + 9\mathbf{s}_{13} + 26\mathbf{m}$	$14\mathbf{m}_{13} + 9\mathbf{s}_{13}$
	dadd	$8\mathbf{m}_{13} + 7\mathbf{s}_{13} + 26\mathbf{m}$	$23\mathbf{m}_{13} + 8\mathbf{s}_{13}$

From Table 1, we can see that the double-and-add ladder is preferred in the situation where there are many consecutive zeros appearing during the iteration. As mentioned in Section 2.4, we can combine the cubical and double-and-add ladder together to

achieve the minimum cost for the biextension exponentiation in practice. The corresponding relationships between the cost of multiplications and squarings over each extension field $\mathbb{F}_{p^k}$ $(k > 1)$ and those over the base field $\mathbb{F}_p$ are illustrated in Table 3.

Table 3 Computational costs of multiplication and squaring in the finite field $\mathbb{F}_{p^k}$ ([2, Table 9] and [9, Table 7]).

k	$\mathbf{m}_k$	$\mathbf{s}_k$
1	$\mathbf{m}$	$\mathbf{s}$
2	$3\mathbf{m}$	$2\mathbf{m}$
4	$9\mathbf{m}$	$2\mathbf{m}_2 = 6\mathbf{m}$
6	$18\mathbf{m}$	$2\mathbf{m}_2 + 3\mathbf{s}_2 = 12\mathbf{m}$
7	$24\mathbf{m}$	$24\mathbf{s}$
8	$27\mathbf{m}$	$2\mathbf{m}_4 = 18\mathbf{m}$
12	$54\mathbf{m}$	$2\mathbf{m}_6 = 36\mathbf{m}$
13	$66\mathbf{m}$	$66\mathbf{s}$
14	$3\mathbf{m}_7 = 72\mathbf{m}$	$2\mathbf{m}_7 = 48\mathbf{m}$
16	$81\mathbf{m}$	$2\mathbf{m}_8 = 54\mathbf{m}$

By taking $\mathbf{s} = \mathbf{m}$ in Table 3, we are able to estimate the computational cost required for each iteration within the biextension computation. The corresponding cost comparison measured by $\mathbb{F}_p$-multiplications between employing Miller's algorithm and biextension on families BLS12, AFG16, BW14 and BW13 is presented in Table 4. As for the computational cost of exploiting the Miller's algorithm, we refer to [2, Table 7] and [19, Table 7] for estimation.

Table 4 The comparison of the corresponding costs of a basic iteration in Miller loop measured by $\mathbb{F}_p$-multiplications between employing Miller's algorithm and biextension (including cubical and double-and-add ladder) on families BLS12, AFG16, BW14 and BW13. Among them, the scenarios in which the biextension computation is proved to be more efficient are marked in red.

Family	Approach	bit = 0	bit = 1
BLS12, $D = 3$	biextension (cubical)	$236\mathbf{m}$	$290\mathbf{m}$
	biextension (dadd)	$214\mathbf{m}$	$688\mathbf{m}$
	Miller	$99\mathbf{m}$	$170\mathbf{m}$
AFG16, $D = 1$ [19]	biextension (cubical)	$300\mathbf{m}$	$381\mathbf{m}$
	biextension (dadd)	$252\mathbf{m}$	$900\mathbf{m}$
	Miller	$200\mathbf{m}$	$382\mathbf{m}$
BW14, $D = 1$ [9]	biextension (cubical)	$744\mathbf{m}$	$888\mathbf{m}$
	biextension (dadd)	$600\mathbf{m}$	$1392\mathbf{m}$
	Miller	$480\mathbf{m}$	$954\mathbf{m}$
BW13, $D = 1$ [19]	biextension (cubical)	$1412\mathbf{m}$	$1518\mathbf{m}$
	biextension (dadd)	$1016\mathbf{m}$	$2046\mathbf{m}$
	Miller	$1636\mathbf{m}$	$3220\mathbf{m}$

It follows from Table 4 that for the majority of situations, computing pairings by utilizing biextension is less efficient than the Miller's algorithm. Nevertheless, for some specific cases, particularly where the embedding degree is an odd prime and the CM

discriminant is $D = 1$, the computation of pairings by leveraging biextension will be more efficient. Consequently, the utilization of biextension for pairing computation holds practical application potential in certain cryptographic scenarios.

5 Conclusion

In this work, we gave a detailed framework for applying biextension to pairing-based cryptography. In particular, we have shown that biextensions are particularly well suited to study and construct pairings on elliptic curves. The theory of biextension is also expected to find other applications in public key cryptography.

Then we have looked at formulas for the biextension arithmetic, which allows to compute the Tate pairing and its variants in practice. These formulas depend on the way biextension elements are represented. In the pairing based literature, it is the *Miller representation* that is (implicitly) used. Instead, in this paper we have looked at the *cubical representation*.

Overall, the efficiency of computing pairings via the cubical representation of biextension is somewhat comparable, but in general slower, to that of the Miller's algorithm. In some specific cases, utilizing the cubical representation is even more efficient. Moreover, compared to the Miller's algorithm, cubical arithmetic is also more suitable for parallel computing. We expect that upon further optimization of the cubical algorithm, it will emerge as a competitive alternative to the Miller algorithm. Indeed, Miller's algorithm had years of optimizations, and the pairing families used in the literature are optimized for this algorithm. But the profile performance of the cubical representation is very different, notably it behaves pretty well for odd-prime embedding degree. We hope that new pairing friendly curves optimized for the cubical arithmetic will emerge as worthy contender. Another useful feature of the cubical representation is that it only needs the x-coordinate of the points $P, Q, P+Q$ to compute $e(P, Q)$, which may prove useful in some cryptographic protocols.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 12441107), Guangdong Major Project of Basic and Applied Basic Research (No. 2019B030302008), and Guangdong Provincial Key Laboratory of Information Security Technology (No. 2023B1212060026), and PEPR PQ-TLS (the France 2030 program under grant agreement ANR-22-PETQ-0008 PQ-TLS).

References

[1] Diego F Aranha, Youssef El Housni, and Aurore Guillevic. A survey of elliptic curves for proof systems. *Designs, Codes and Cryptography*, 91(11):3333–3378, 2023.

[2] Diego F. Aranha, Georgios Fotiadis, and Aurore Guillevic. A short-list of pairing-friendly curves resistant to the Special TNFS algorithm at the 192-bit security level. *IACR Communications in Cryptology*, 1(3):44, October 2024.

[3] Paulo SLM Barreto, Steven D Galbraith, Colm Ó' hÉigeartaigh, and Michael Scott. Efficient pairing computation on supersingular abelian varieties. *Designs, Codes and Cryptography*, 42:239–271, 2007.

[4] Paulo SLM Barreto, Hae Y Kim, Ben Lynn, and Michael Scott. Efficient algorithms for pairing-based cryptosystems. In *Advances in Cryptology—CRYPTO 2002: 22nd Annual International Cryptology Conference Santa Barbara, California, USA, August 18–22, 2002 Proceedings 22*, pages 354–369. Springer, 2002.

[5] Dan Boneh and Matt Franklin. Identity-Based Encryption from the Weil Pairing. In Joe Kilian, editor, *Advances in Cryptology — CRYPTO 2001*, pages 213–229, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[6] Dan Boneh, Ben Lynn, and Hovav Shacham. Short Signatures from the Weil Pairing. In Colin Boyd, editor, *Advances in Cryptology — ASIACRYPT 2001*, pages 514–532, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[7] Shiping Cai, Zhi Hu, Zheng-An Yao, and Chang-An Zhao. The elliptic net algorithm revisited. *Journal of Cryptographic Engineering*, 14(1):43–55, 2024.

[8] Binglong Chen and Chang-An Zhao. An improvement of the elliptic net algorithm. *IEEE Transactions on Computers*, 65(9):2903–2909, 2015.

[9] Yu Dai, Debiao He, Cong Peng, Zhijian Yang, and Chang-an Zhao. Revisiting Pairing-Friendly Curves with Embedding Degrees 10 and 14. In Kai-Min Chung and Yu Sasaki, editors, *Advances in Cryptology – ASIACRYPT 2024*, pages 454–485, Singapore, 2025. Springer Nature Singapore.

[10] Yu Dai, Fangguo Zhang, and Chang-an Zhao. Don't Forget Pairing-Friendly Curves with Odd Prime Embedding Degrees. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2023(4):393–419, Aug. 2023.

[11] Iwan Duursma and Hyang-Sook Lee. Tate Pairing Implementation for Hyperelliptic Curves $y^2 = x^p - x + d$. In Chi-Sung Laih, editor, *Advances in Cryptology - ASIACRYPT 2003*, pages 111–123, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[12] Youssef El Housni and Aurore Guillevic. Optimized and Secure Pairing-Friendly Elliptic Curves Suitable for One Layer Proof Composition. In Stephan Krenn, Haya Shulman, and Serge Vaudenay, editors, *Cryptology and Network Security*, pages 259–279, Cham, 2020. Springer International Publishing.

[13] Youssef El Housni and Aurore Guillevic. Families of SNARK-Friendly 2-Chains of Elliptic Curves. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology – EUROCRYPT 2022*, pages 367–396, Cham, 2022. Springer International Publishing.

[14] Emmanuel Fouotsa, Laurian Azebaze Guimagang, and Raoul Ayissi. x-superoptimal pairings on elliptic curves with odd prime embedding degrees: BW 13-P 310 and BW 19-P 286. *Applicable Algebra in Engineering, Communication and Computing*, pages 1–19, 2023.

[15] David Freeman, Michael Scott, and Edlyn Teske. A taxonomy of pairing-friendly elliptic curves. *Journal of cryptology*, 23:224–280, 2010.

[16] Steven D Galbraith, Keith Harrison, and David Soldera. Implementing the Tate pairing. In *International Algorithmic Number Theory Symposium*, pages 324–337.

Springer, 2002.
[17] Robert P. Gallant, Robert J. Lambert, and Scott A. Vanstone. Faster Point Multiplication on Elliptic Curves with Efficient Endomorphisms. In Joe Kilian, editor, *Advances in Cryptology — CRYPTO 2001*, pages 190–200, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.
[18] Alexandre Grothendieck. *Groupes de Monodromie en Géométrie Algébrique (SGA 7)*, volume 288 of *Lecture Notes in Mathematics*. Springer-Verlag, 1972.
[19] Aurore Guillevic. A Short-List of Pairing-Friendly Curves Resistant to Special TNFS at the 128-Bit Security Level. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *Public-Key Cryptography – PKC 2020*, pages 535–564, Cham, 2020. Springer International Publishing.
[20] F. Hess, N.P. Smart, and F. Vercauteren. The Eta Pairing Revisited. *IEEE Transactions on Information Theory*, 52(10):4595–4602, 2006.
[21] Florian Hess. Pairing Lattices. In Steven D. Galbraith and Kenneth G. Paterson, editors, *Pairing-Based Cryptography – Pairing 2008*, pages 18–38, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.
[22] Eunjeong Lee, Hyang-Sook Lee, and Cheol-Min Park. Efficient and Generalized Pairing Computation on Abelian Varieties. *IEEE Transactions on Information Theory*, 55(4):1793–1803, 2009.
[23] Jianming Lin, Chang-An Zhao, and Yuhao Zheng. Efficient Implementation of Super-optimal Pairings on Curves with Small Prime Fields at the 192-bit Security Level. Cryptology ePrint Archive, Paper 2024/1195, 2024.
[24] David Lubicz and Damien Robert. Arithmetic on Abelian and Kummer Varieties. *Finite Fields and Their Applications*, 39:130–158, 5 2016.
[25] Seiichi Matsuda, Naoki Kanayama, Florian Hess, and Eiji Okamoto. Optimised Versions of the Ate and Twisted Ate Pairings. In Steven D. Galbraith, editor, *Cryptography and Coding*, pages 302–312, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.
[26] Victor S Miller. The Weil pairing, and its efficient calculation. *Journal of cryptology*, 17:235–261, 2004.
[27] Hermann Minkowski. *Geometrie der zahlen*. BG Teubner, 1910.
[28] David Mumford. Bi-extensions of formal groups. *Algebraic geometry*, (307-322), 1969.
[29] Yan Feng Qi, Chun Ming Tang, Baoan Guo, and Mao Zhi Xu. Super-optimal pairings. *Applied Mechanics and Materials*, 281:127–133, 2013.
[30] Damien Robert. Fast pairings via biextensions and cubical arithmetic. Cryptology ePrint Archive, Paper 2024/517, 2024.
[31] Michael Scott. Faster Pairings Using an Elliptic Curve with an Efficient Endomorphism. In Subhamoy Maitra, C. E. Veni Madhavan, and Ramarathnam Venkatesan, editors, *Progress in Cryptology - INDOCRYPT 2005*, pages 258–269, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.
[32] Katherine Stange. *Elliptic nets and elliptic curves*. PhD thesis, Brown University, 2008.
[33] Katherine E Stange. The Tate pairing via elliptic nets. In *Pairing-Based Cryptography–Pairing 2007: First International Conference, Tokyo, Japan, July*

2-4, 2007. Proceedings 1, pages 329–348. Springer, 2007.
[34] F. Vercauteren. Optimal Pairings. Cryptology ePrint Archive, Paper 2008/096, 2008.
[35] Frederik Vercauteren. Optimal Pairings. *IEEE Transactions on Information Theory*, 56(1):455–461, Jan 2010.
[36] Lawrence C Washington. *Elliptic curves: number theory and cryptography*. Chapman and Hall/CRC, 2008.
[37] Chang-An Zhao, Fangguo Zhang, and Jiwu Huang. A note on the Ate pairing. *International Journal of Information Security*, 7(6):379–382, 2008.

Appendix A The related algorithms

In this appendix, we present some associated algorithms required in the pairing computation through biextension, including x-only cubical point doubling, differential addition and compatible addition algorithms on Kummer line $K = E/\langle \pm 1 \rangle$ over $\mathbb{F}_{p^k}$, with $j(E) = 0$ or $j(E) = 1728$.

Algorithm 7 x-only cubical point doubling on the curve $E : y^2 = x^3 + b$

Input: A point $P = (X_P : Z_P)$ in $E(\mathbb{F}_{p^k})$.
Output: The coordinates $(X_{[2]P} : Z_{[2]P})$ of the double of P.

1: $t_1 \leftarrow X_P^2$
2: $t_2 \leftarrow t_1 \cdot X_P$
3: $t_3 \leftarrow Z_P^2$
4: $t_4 \leftarrow t_3 \cdot Z_P$
5: $t_5 \leftarrow t_2 - 2 \cdot 4b \cdot t_4$
6: $t_6 \leftarrow 4 \cdot t_2 + 4b \cdot t_4$
7: $X_{[2]P} \leftarrow X_P \cdot t_5$
8: $Z_{[2]P} \leftarrow Z_P \cdot t_6$
9: **return** $X_{[2]P},\ Z_{[2]P}$ ▷ Total cost: $4\mathbf{m}_k + 2\mathbf{s}_k + 1\mathbf{m}_0$

Algorithm 8 x-only cubical differential addition on the curve $E : y^2 = x^3 + b$

Input: The points $P = (X_P : Z_P), Q = (X_Q : Z_Q)$ and $P - Q = (X_{P-Q} : Z_{P-Q}) \in E(\mathbb{F}_{p^k})/\langle \pm 1 \rangle$ with $Z_{P-Q} = 1$. The inverse of the X-coordinate of the differential of P and Q: iX_{P-Q}.

Output: The coordinate $(X_{P+Q} : Z_{P+Q})$

1: $t_1 \leftarrow X_P + Z_P$
2: $t_2 \leftarrow X_P - Z_P$
3: $t_3 \leftarrow X_Q + Z_Q$
4: $t_4 \leftarrow X_P \cdot X_Q$
5: $t_5 \leftarrow Z_P \cdot Z_Q$
6: $t_6 \leftarrow t_1 \cdot t_3 - t_4 - t_5$
7: $t_7 \leftarrow t_2 \cdot t_3 - t_4 + t_5$
8: $X_{P+Q} \leftarrow (-4b \cdot t_5 \cdot t_6 + t_4^2)$
9: $Z_{P+Q} \leftarrow t_7^2 \cdot X_{P-Q}$
10: **return** $X_{P+Q},\ Z_{P+Q}$ ▷ Total cost: $6\mathbf{m}_k + 2\mathbf{s}_k + 1\mathbf{m}_0$

Algorithm 9 x-only cubical point doubling on the curve $E : y^2 = x^3 + ax$

Input: A point $P = (X_P : Z_P)$ in $E(\mathbb{F}_{p^k})$.

Output: The coordinates $(X_{[2]P} : Z_{[2]P})$ of the double of P.

1: $t_1 \leftarrow X_P^2$
2: $t_2 \leftarrow Z_P^2$
3: $t_3 \leftarrow a \cdot t_2$
4: $X_{[2]P} \leftarrow (t_1 - t_3)^2$
5: $t_4 \leftarrow 4X_P \cdot Z_P$
6: $Z_{[2]P} \leftarrow t_4 \cdot (t_1 + t_3)$
7: **return** $X_{[2]P},\ Z_{[2]P}$ ▷ Total cost: $2\mathbf{m}_k + 3\mathbf{s}_k + 1\mathbf{m}_0$

Algorithm 10 x-only cubical differential addition on the curve $E : y^2 = x^3 + ax$

Input: Two points $P = (X_P : Z_P), Q = (X_Q : Z_Q) \in E(\mathbb{F}_{p^k})$ with $Z_{P-Q} = 1$. The inverse of the X-coordinate of the differential of P and Q: iX_{P-Q}.

Output: The coordinate $(X_{P+Q} : Z_{P+Q})$

1: $t_1 \leftarrow X_P \cdot Z_Q$
2: $t_2 \leftarrow X_Q \cdot Z_P$
3: $t_3 \leftarrow (X_P + Z_P) \cdot (X_Q - a \cdot Z_Q) - t_2 + a \cdot t_1$
4: $t_4 \leftarrow t_3^2$
5: $t_5 \leftarrow (t_1 - t_2)^2$
6: $X_{P+Q} \leftarrow t_4 \cdot iX_{P-Q}$
7: $Z_{P+Q} \leftarrow t_5$
8: **return** $X_{P+Q},\ Z_{P+Q}$ ▷ Total cost: $4\mathbf{m}_k + 2\mathbf{s}_k + 2\mathbf{m}_0$

Algorithm 11 Compatible addition on the curve $E : y^2 = x^3 + b$

Input: Four points $P_1 = (X_{P_1} : Z_{P_1})$, $P_2 = (X_{P_2} : Z_{P_2})$, $P_1 + Q = (X_{P_1+Q} : Z_{P_1+Q})$, $P_2 - Q = (X_{P_2-Q} : Z_{P_2-Q}) \in E(\mathbb{F}_{p^k})$ with $Z_{P_2} = Z_{P_2-Q} = 1$.
Output: The coordinate $(X_{P_1+P_2} : Z_{P_1+P_2})$

1: $t_1 \leftarrow (X_{P_1} - X_{P_2} \cdot Z_{P_1})^2$
2: $t_2 \leftarrow X_{P_2} \cdot Z_{P_1} + X_{P_1}$
3: $t_3 \leftarrow X_{P_1} \cdot X_{P_2}$
4: $t_4 \leftarrow -4b \cdot Z_{P_1} \cdot t_2 + t_3^2$
5: $t_5 \leftarrow 2(2b \cdot Z_{P_1}^2 + t_2 \cdot t_3)$
6: $t_6 \leftarrow (X_{P_1+Q} - X_{P_2-Q} \cdot Z_{P_1+Q})^2$
7: $t_7 \leftarrow X_{P_2-Q} \cdot Z_{P_1+Q} + X_{P_1+Q}$
8: $t_8 \leftarrow X_{P_1+Q} \cdot X_{P_2-Q}$
9: $t_9 \leftarrow -4b \cdot Z_{P_1+Q} \cdot t_7 + t_8^2$
10: $t_{10} \leftarrow 2(2b \cdot Z_{P_1+Q}^2 + t_7 \cdot t_8)$
11: $X_{P_1+P_2} \leftarrow t_4 \cdot t_{10} - t_5 \cdot t_9$
12: $Z_{P_1+P_2} \leftarrow t_4 \cdot t_6 - t_1 \cdot t_9$
13: **return** $X_{P_1+P_2}$, $Z_{P_1+P_2}$ ▷ Total cost: $12\mathbf{m}_k + 6\mathbf{s}_k + 4\mathbf{m}_0$

Algorithm 12 Compatible addition on the curve $E : y^2 = x^3 + ax$

Input: Four points $P_1 = (X_{P_1} : Z_{P_1})$, $P_2 = (X_{P_2} : Z_{P_2})$, $P_1 + Q = (X_{P_1+Q} : Z_{P_1+Q})$, $P_2 - Q = (X_{P_2-Q} : Z_{P_2-Q}) \in E(\mathbb{F}_{p^k})$ with $Z_{P_2} = Z_{P_2-Q} = 1$.
Output: The coordinate $(X_{P_1+P_2} : Z_{P_1+P_2})$

1: $t_1 \leftarrow X_{P_2} \cdot Z_{P_1}$
2: $t_2 \leftarrow X_{P_2} \cdot X_{P_1}$
3: $t_3 \leftarrow X_{P_2-Q} \cdot Z_{P_1+Q}$
4: $t_4 \leftarrow X_{P_2-Q} \cdot X_{P_1+Q}$
5: $t_5 \leftarrow (t_2 - a \cdot Z_{P_1})^2$
6: $t_6 \leftarrow (t_4 - a \cdot Z_{P_1+Q})^2$
7: $t_7 \leftarrow 2(t_3 + X_{P_1+Q}) \cdot (t_4 + a \cdot Z_{P_1+Q}) \cdot t_5$
8: $t_8 \leftarrow 2(t_1 + X_{P_1}) \cdot (t_2 + a \cdot Z_{P_1}) \cdot t_6$
9: $X_{P_1+P_2} \leftarrow t_7 - t_8$
10: $t_9 \leftarrow (t_3 - X_{P_1+Q}) \cdot (t_2 - a \cdot Z_{P_1})$
11: $t_{10} \leftarrow (t_1 - X_{P_1}) \cdot (t_4 - a \cdot Z_{P_1+Q})$
12: $Z_{P_1+P_2} \leftarrow (t_9 + t_{10}) \cdot (t_9 - t_{10})$
13: **return** $X_{P_1+P_2}$, $Z_{P_1+P_2}$ ▷ Total cost: $11\mathbf{m}_k + 2\mathbf{s}_k + 4\mathbf{m}_0$