# Biextension 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 Key Laboratory of Information Security, 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 play a significant role in modern public-key cryptography, with the improvement of Tate pairings and their variants representing a crucial area of cryptographic research. Currently, the Miller's algorithm stands as the most widely adopted and efficient method for pairing computation. In this paper, we revisit the application of the technique of biextension for pairing computation and extend it to 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 biextension-based methods with the Miller's algorithm on various well-known families of pairing-friendly elliptic curves. Our results demonstrate that the biextension-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  super-optimal ate pairing  Miller's algorithm  biextension  cubical ladder

# 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 [?], short signatures [?], and zero-knowledge proofs [?, ?, ?]. 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 the majority of scenarios, the Tate pairing demonstrates greater efficiency. Consequently, a substantial amount of research have focused on optimizing the Tate pairing. One of the research objectives is to shorten the length of the Miller loop. Duursma and Lee [?], along with Barreto *et al.* [?] have successfully shortened the length of the Miller iteration required for the Tate pairing on supersingular abelian varieties leveraging the $\eta_T$ method. In 2006, Hess *et al.* [?] 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 [?, ?, ?] 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 $\varphi$ the Euler function. In certain specific cases where a curve possesses efficiently-computable endomorphisms distinct from powers of the Frobenius, the iteration length can be shortened to $\log_2 r/2\varphi(k)$ and the corresponding pairing is named **super-optimal (ate) pairing**. More recently, there are many studies [?, ?, ?, ?, ?] targeting on the computation of super-optimal ate pairing.

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 [?]. Since then, a huge number of works [?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?] 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 [?] is another method for computing the pairings. In 2015, Chen *et al.* [?] 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.* [?] 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 [?] presented a novel approach by leveraging 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 [?] 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 biextension for pairing computations in elliptic curve cryptography (ECC) and make concrete cost analysis.

### 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. By employing the twisting isomorphism, we have derived more precise formulas and algorithms for the ate pairing and the optimal ate pairing computation through biextension compared to those presented in [**?**]. Furthermore, by combining biextension with the efficiently-computable endomorphisms, we propose efficient formulas for the super-optimal ate pairing. In addition, we present an optimized shared cubical ladder algorithm for the implementation.
2. We conduct a meticulous efficiency analysis for the algorithms in this paper. Subsequently, we compared the efficiency of our method with that of the Miller's algorithm on several well-known families of pairing-friendly curves. The results illustrate that the biextension 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 odd and the CM discriminant is 1), making it a 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 corresponding mathematical preliminaries and fundamental descriptions required in this paper.

Let $E$ denote an ordinary elliptic curve defined over a finite field $\mathbb{F}_p$, where $p$ is a prime satisfying $p > 5$. If $E$ is a short Weierstrass curve, then the rational points $(x, y)$ with $x, y \in \mathbb{F}_p$ in the group $E(\mathbb{F}_p)$ 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 [?, 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 the order 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 definitions of the three pairing subgroups $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ with order $r$ are presented as follows

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

where $E[r] = \{P \in E \mid [r]P = \mathcal{O}_E\}$ and $\mu_r$ 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 $e = k/d$ [?]. 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 $\mathbb{F}_{p^k}$, but not over $\mathbb{F}_{p^e}$. According to [?, Proposition 1], all twists corresponding to $D \in \mathbb{F}_{p^e}^*/(\mathbb{F}_{p^e}^*)^d$ are given by

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

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 $E'$

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

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 [?, 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 $\mathbb{Z} \subsetneq R \subseteq \mathcal{O}_K$ [?]. 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 $\sigma$ be an endomorphism of $E$ over $\mathbb{F}_p$. Then $\sigma$ can be conveniently represented as $\sigma = a + b\sqrt{-D}$, where $a, b \in \mathbb{Q}$ and $2a, 2b \in \mathbb{Z}$. If $\sigma$ is an $n$-isogeny (a degree-$n$ endomorphism), it satisfies the following characteristic equation

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

as the reduced norm of $\sigma$ is given by $\mathrm{Nrd}(\sigma) = n$. These endomorphisms allow for fast scalar multiplications via the GLV method [?]. 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.

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

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

There exists an endomorphism $\sigma : (x, y) \mapsto (\alpha x, y)$ on $E_1$, associated with $\frac{1+\sqrt{-3}}{2}$ in the endomorphism ring $\mathrm{End}_p(E_1)$, where $\alpha$ 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 endomorphism is $\sigma : (x, y) \mapsto (-x, \beta y)$, associated with $\pm\sqrt{-1}$ in $\mathrm{End}_p(E_2)$, where $\beta$ is a primitive fourth root of unity in $\mathbb{F}_p^*$. This endomorphism 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 definitions of the Miller function and Miller's algorithm.

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 [?].

### 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 [?], is an optimized variant of the Tate pairing on $\mathbb{G}_2 \times \mathbb{G}_1$ and achieves a shorter Miller loop by employing the $p$-power Frobenius endomorphism $\pi$. Let $P$ and $Q$ be two points in $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. Denote by $\lambda$ 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 [?, ?] have sought to further shorten the length of the Miller loop through multiplying or dividing the ate pairings.

Vercauteren [?] 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 [?], 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 [?] on $E$ is defined as follows

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

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** [?, ?, ?]. Such pairings [?, ?, ?, ?, ?, ?, ?] are constructed by compositing the power of Frobenius endomorphism and the GLV-endomorphism.

## 2.3 Biextensions

Biextensions were first introduced by Mumford in [?]. As mentioned in [?], 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 $\mathrm{X}_D$, can be defined as follows.

**Definition 1** ([?]). *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 $\mathrm{X}_D$ simply as $g_{P,Q} \in \mathrm{X}_D$. The biextension $\mathrm{X}_D$ 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

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

These definitions ensure that the group laws respect the structure of the biextension and allow for a rich arithmetic framework. Since $\mathrm{X}_D$ is a symmetric function, we also have

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

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

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

$$g_{P,Q}^{\star_1,-1} = g_{-P,Q} = \frac{1}{g_{P,Q}} \cdot \frac{g_{-P,P}}{g_{-P,P}(\cdot + Q)}. \tag{4}$$

We note that the RHS does not depend on the choice of representative for $g_{-P,P}$.

As per Definition 1 and Eq. (4), 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 [?, Porism 3.10].

**Lemma 1** ([**?**])**.** *Let $g_{P,Q} \in X_D$ and suppose that $P \in E[r]$. Then, the Miller function $f_{r,-P}$ operates on the cycle $(\cdot) - (\cdot + Q)$ as given by $\frac{g_{[r]P,Q}(\cdot)}{g^r_{P,Q}(\cdot)}$.*

From the point of view of 1, as explained by Grothendieck in [**?**], 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.

In this paper, for the function $g_{P,Q}$, we will look at the cubical representation as outlined in [**?**, 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) represented by the cubical coordinate $Z_1(\widetilde{P})$. The biextension function $g_{P,Q}$ is then represented as a quotient of cubical functions

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

where $Z_1$ is a choice of section of the divisor $D = (0_E)$.

Here, the point $Z_1(\widetilde{R+P+Q})$ is evaluated via the cubical arithmetic, using the cube $0, P, Q, R, Q+R, P+R, P+Q, P+Q+R$:

$$\frac{Z_1(\widetilde{P+Q+R})Z_1(\widetilde{P})Z_1(\widetilde{Q})Z_1(\widetilde{R})}{Z_1(\widetilde{0_E})Z_1(\widetilde{Q+R})Z_1(\widetilde{P+R})Z_1(\widetilde{P+Q})} = g_{P,Q}(R)/g_{P,Q}(0_E).$$

We remark that the RHS does not depend on the choice of biextension function $g_{P,Q}$ above $(P,Q)$.

For our algorithms, it will be convenient to switch to cubical points of level 2. We let $Z = Z_1^2$, this is a section of $2D = 2(0_E)$, and $X$ another section such that $x = X/Z$. A level 2 cubical point $\widetilde{P}$ is then determined by $\widetilde{P} = (X(\widetilde{P}), Z(\widetilde{P})))$; for ease of notations we will drop the tilde and use the notations $X_P, Z_P$. Working with level 2 cubical points mean 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.

If $R = \mathcal{O}_E$, direct evaluation of $g_{P,Q}$ at $R$ is inadvisable since the point at infinity constitutes a zero of $g_{P,Q}$. According to [**?**, Remark 2.8], an extended value is required for this special case. By defining the uniformizer as $u_{\mathcal{O}_E} = \frac{Z}{X}$, we can express $Z$ as $u_{\mathcal{O}_E} \cdot X$. Therefore, the extended value of $Z$ at $\mathcal{O}_E$ becomes $X_{\mathcal{O}_E}$, leading to the evaluation of $g_{P,Q}$ at $\mathcal{O}_E$ being

$$g_{P,Q}(\mathcal{O}_E) = \frac{Z_{P+Q} \cdot X_{\mathcal{O}_E}}{Z_P \cdot Z_Q}. \tag{5}$$

Hereafter, we will simply denote $g_{P,Q}(\mathcal{O}_E)$ as $g_{P,Q}$. On this basis, if $P$ is an $r$-torsion point, the evaluation of $Z$ at $[r]P$ yields $X_{[r]P}$. Thus, it follows that

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

Without loss of generality, the projective $x$-coordinate of $\mathcal{O}_E$ is taken as $X_{\mathcal{O}_E} = 1$, and we will typically drop it from our notation. Given that $\frac{X_{[r]P+Q}}{X_Q} = \frac{Z_{[r]P+Q}}{Z_Q}$, the function $g_{[r]P,Q}$ can be expressed alternatively as

$$g_{[r]P,Q} = \frac{X_{[r]P+Q}}{X_{[r]P} \cdot X_Q}.$$

The coordinate values $X_{[r]P+Q}$ and $X_{[r]P}$ can be efficiently computed using the cubical ladder algorithm described in [**?**, Algorithm 4.2].

## 2.4 Biextension arithmetic

In this section, we look more closely at the biextension, which we will use for pairings. We work in level 1 for now, going to level 2 simply involves taking squares everywhere.

First, a biextension element $g_{P,Q}$ has for 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 normalised at $\mathcal{O}_E$ with respect to the uniformiser $x/y$. In the special case where $Q = -P$, we instead take $g_{P,-P} = 1/v_P$.

Therefore, Eq. (4) 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

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

We can also rewrite lemma 1 as

$$f_{r,P}((\cdot + Q) - (\cdot)) = \frac{g_{[r]P,Q}(\cdot)}{g^r_{P,Q}(\cdot)} \cdot \frac{v^r_P}{v_{[r]P}}((\cdot + Q) - (\cdot)). \tag{6}$$

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

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

As explained in section 2.1 we will use twists, hence 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 of elliptic curves. Then $g_{P,Q} \mapsto g_{\phi^{-1}(P),\phi^{-1}(Q)} = \phi^* g_{P,Q} \coloneqq g_{P,Q} \circ \phi$ is an isomorphism from the biextension on $E_2$ associated to $(0_{E_1})$ to the biextension on $E_1$ associated to $(0_{E_1})$.*

*Proof.* This follows from the functoriality of biextensions. This can also be directly seen as follows: let $P' = \phi^{-1}(P), Q' = \phi^{-1}(Q)$, then $\phi^* g_{P,Q}$ has for divisor $(-P' - Q') + (0_{E_1}) - (P') - (Q')$ so is a biextension element above $(P', Q')$. Furthermore it is immediate from their definition that $\phi^*$ is compatible with the biextension laws $\star_1$ and $\star_2$. □

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

## 2.5 Cubical arithmetic for biextensions

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

Suppose that we have a cubical representation of the biextension elements $g_{P_1,Q} = [\widetilde{P_1}, \widetilde{Q}, \widetilde{0}_E, \widetilde{P_1 + Q}]$, $g_{P_2,Q} = [\widetilde{P_2}, \widetilde{Q}, \widetilde{0}_E, \widetilde{P_2 + Q}]$, $g_{P_1-P_2,Q} = [\widetilde{P_1 - P_2}, \widetilde{Q}, \widetilde{0}_E, \widetilde{P_1 - P_2 + Q}]$, then we can compute $g_{P_1+P_2,Q} = [\widetilde{P_1 + P_2}, \widetilde{Q}, \widetilde{0}_E, \widetilde{P_1 + P_2 + Q}]$, via $\widetilde{P_1 + P_2} = \mathrm{xAdd}(\widetilde{P_1}, \widetilde{P_2}, \widetilde{P_1 - P_2})$ and $\widetilde{P_1 + P_2 + Q} = \mathrm{xAdd}(\widetilde{P_1 + Q}, \widetilde{P_2}, \widetilde{P_1 - P_2 + Q})$ where xAdd denotes a cubical differential addition. We refer to section A for explicit formulas for cubical points of level 2.

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 $\widetilde{P_1 + P_2 + Q}$, 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 $g_{P,Q} \mapsto g_{rP,Q}$. Either we use a cubical ladder, computing $n\widetilde{P}, (n+1)\widetilde{P}, n\widetilde{P}+\widetilde{Q}$ at each step. The ladder uses one cubical doubling and two cubical differential additions at each step.

Or we use a double and add approach, keeping only $n\widetilde{P}, n\widetilde{P}+\widetilde{Q}$ at each step. When the current bit is 0, we do a biextension doubling by computing $2n\widetilde{P}, 2n\widetilde{P}+\widetilde{Q}$, this costs one cubical doubling and one cubical differential addition. When the current bit is 1, we first recover $(n+1)P = \mathrm{cAdd}(nP, P, nP+Q, P-Q)$ 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 [**?**], and we refer to section A for explicit formulas. 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.

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

**Proposition 2.** *Let $\phi : E_1 \to E_2$ be an isomorphism of elliptic curves. Let $\widetilde{\phi}$ be the unique lift of $\phi$ to cubical points that sends $\widetilde{0}_{E_1}$ to $\widetilde{0}_{E_2}$. 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 $\phi$. Then given a cube $0, P, Q, R, Q+R, P+R, P+Q, P+Q+R$ in $E_2$, so that we have:

$$\frac{Z_2(\widetilde{P+Q+R})Z_2(\widetilde{P})Z_2(\widetilde{Q})Z_2(\widetilde{R})}{Z_2(\widetilde{0_{E_2}})Z_2(\widetilde{Q+R})Z_2(\widetilde{P+R})Z_2(\widetilde{P+Q})} = g_{P,Q}(R)/g_{P,Q}(0_{E_2}),$$

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

$$\frac{Z_1(\widetilde{P'+Q'+R'})Z_1(\widetilde{P'})Z_1(\widetilde{Q'})Z_1(\widetilde{R'})}{Z_1(\widetilde{\phi}^{-1}(\widetilde{0_{E_2}}))Z_1(\widetilde{Q'+R'})Z_1(\widetilde{P'+R'})Z_1(\widetilde{P'+Q'})} = g_{P',Q'}(R')/g_{P',Q'}(0_{E_1}).$$

And so the two cubical laws are compatible as long as $\widetilde{\phi}^{-1}(\widetilde{0_{E_2}}) = \widetilde{0_{E_1}}$. □

**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 $\widetilde{0}_E = (1,0)$, we have that $\widetilde{\phi}(X,Z) = (X, Z/\xi^2)$ is a cubical isomorphism.*

# 3 Main Results

In this section, we present a comprehensive framework to derive precise formulas for the ate pairing, optimal ate pairing, and super-optimal ate pairing by leveraging the technique of biextensions. For each type of pairing, we delineate the corresponding computational procedures and provide illustrative examples. Assume that we operate on the Kummer line $K = E/\langle \pm 1 \rangle$ of an elliptic curve $E$ associated with the biextension $\mathrm{X}_{2(\mathcal{O}_E)}$ with sections $(X, Z)$ throughout the remainder of this paper.

## 3.1 Biextension for the Tate pairing

As a warm up, we first look at the Tate pairing and on how to exploit twisting isomorphisms.

Let $P \in E(\mathbb{F}_{p^k})[r]$ and $Q \in E(\mathbb{F}_{p^k})$. Then 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 $R \in E(\mathbb{F}_{p^k})$. By lemma 1, we have (up to a sign): $e_r(P,Q) = \frac{g_{[r]P,Q}}{g^r_{P,Q}}(R)^{(p^k-1)/r} = g_{[r]P,Q}(R)^{(p^k-1)/r}$, where the last equality assumes that $g_{P,Q}$ is chosen to be rational (e.g., the one normalised at $0_E$). Since $[r]P = 0_E$, $g_{[r]P,Q}$ is actually a constant function. So we have $g_{[r]P,Q}(R)^{(p^k-1)/r} = g_{[p^k-1]P,Q}(R)$.

There is a more intrinsic reformulation that does not depend on any 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. Then observe that $g_{[q]P,Q}$ and $\pi_q \cdot g_{P,Q}$ have both the same divisor, hence they differ by a constant $c$. Furthermore, this constant does not depends on the choice of representative for $g_{P,Q}$, even non rational. So we may assume that $g_{P,Q}$ is rational to determine $c$, and the computation above shows that $g_{[q]P,Q}(R) = c g_{P,Q}(R) = (g_{[p^k-1]P,Q} \star_1 g_{P,Q})(R) = e_r(P,Q) g_{P,Q})(R)$. So $c = e_r(P,Q)$. In summary:

$$e_r(P,Q) = \frac{g_{[q]P,Q}}{\pi_q \cdot g_{P,Q}} \tag{7}$$

Now let $\phi : E' \to E$ be a twisting isomorphism. Then by proposition 1, we have $g_{[q]P',Q'} = \phi^* g_{[q]P,Q}$, so we can work on $E'$ to compute the biextension exponentiation.

One needs to be careful that in general, even if we start with a rational (over $\mathbb{F}_{p^k}$) biextension function $g_{P,Q}$, then $g_{P',Q'}$ may not be rational. Reformulating: if we compute the constant function $g_{[r]P',Q'}$ on $E'$, starting with $g_{P',Q'}$ normalised at $0_{E'}$ for ease of computation, then $g_{P',Q'} = \phi^* g_{P,Q}$ for some $g_{P,Q}$ that may not be rational. So 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. (7) to adjust the result to get the correct Tate pairing. However, in our applications, we will always use a twisting isomorphism $\phi$ that is defined over $\mathbb{F}_{p^k}$ rather than an extension, so the problem is moot.

In fact, we will see a similar phenomena for the other pairings we consider: the twists we use are actually always in a strict subfield of $\mathbb{F}_{p^k}$, and so the adjusting factor when going to the twist will always be killed by the final exponentiation by $(p^k-1)/r$.

One last remark: we can use eq. (7) to relate the twisting correcting factor with the automorphism $\alpha$ inducing the twist $E'$, i.e. such that $\phi\pi\phi^{-1} = \alpha\pi'$. Indeed, we have $\phi^* g_{[q]P,Q} = g_{[q]P',Q'}$. However, $\phi^*(\pi_q \cdot g_{P,Q})$ differ from $\pi'_q \cdot g_{P',Q'}$ in general. Indeed unraveling the formulas we get that $\phi^*(\pi_q \cdot g_{P,Q}) = (\pi'_q \cdot g_{P',Q'}) \circ \alpha^{-1}$.

A similar reasoning holds using the cubical arithmetic and proposition 2. The Tate pairing is given by $e_r(P,Q) = \left(\frac{Z(rP+Q)Z(0)}{Z(Q)Z(rP)}\right)^{(q^k-1)/r}$, as long as we start with rational cubical points $\widetilde{P}, \widetilde{Q}, \widetilde{P+Q}$, e.g. normalised to have $Z=1$. But by example 1, if $Z(\widetilde{P}) = 1$, then $Z(\widetilde{P}') \neq 1$. So conversely, if we want to use cubical arithmetic on $E'$, and we start with normalised points $\widetilde{P}', \widetilde{Q}', \widetilde{P+Q}'$ to speed up the cubical arithmetic, it means that going back to $E$ we were doing cubical arithmetic with non normalised points, potentially even non rational cubical points. So we need to adjust by a suitable power of the conversion factor $\xi^2$ in the end. In practice, for the twisting isomorphism we consider, $\xi^2$ lives in a strict subfield, so will be killed by the final exponentiation anyway.

### 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 $\pi$ 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 $\lambda \equiv p \mod r$, we have $\lambda = t-1$, with $t$ the trace of the Frobenius endomorphism.

By setting $Z_P = Z_Q = Z_{P+Q} = 1$, it follows from [?, Section 5.1] and Eq. (5) that the square of the reduced ate pairing can be expressed as

$$a_\lambda(P,Q)^2 = \left(\frac{g_{[t-1]Q,P}}{g^p_{Q,P}}\right)^{\frac{p^k-1}{r}} = \left(\frac{Z_{[t-1]Q+P}}{Z_{[t-1]Q}}\right)^{\frac{p^k-1}{r}}.$$

In practical applications, most of the curves utilized in pairing-based cryptography admit twists. Therefore, it is essential to employ the twisting isomorphism $\phi$ to enhance the efficiency of ate pairing. According to [?], 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.

**Theorem 1.** *With the aforementioned notations, let $P \in \mathbb{G}_1$ and $Q' \in E'(\mathbb{F}_{p^{k/d}})[r]$ such that $Q = \phi(Q') \in \mathbb{G}_2$, where $\phi$ is the degree-$d$ twisting isomorphism. 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*

$$a_b(P, Q) = a_\lambda(P, Q)^2 = Z_{[t-1]Q'+\phi^{-1}(P)}^{\frac{p^k-1}{r}}.$$

*Proof.* The ate pairing is given by the monodromy $g_{[t-1]Q,P} = g_{Q,P}^{\star,t-1} = a \cdot \pi(g_{Q,P})$ [?, Remark 3.22], where the result of the pairing $a$ does not rely on whether $g_{Q,P}$ is rational. On the twist $E'$, we consider the pullback $\phi^*$ and obtain a function

$$\phi^*(g_{Q,P}(\cdot)) = g_{Q,P} \circ \phi(\cdot) = g_{Q',\phi^{-1}(P)}(\cdot) \in \mathrm{X}_{2(\mathcal{O}_{E'})}.$$

The corresponding monodromy is

$$g_{Q',\phi^{-1}(P)}^{\star,t-1} = g_{[t-1]Q',\phi^{-1}(P)} = a' \cdot \pi'(g_{Q',\phi^{-1}(P)}).$$

By $\phi \circ \pi' = \pi$, we have

$$g_{Q,P}^{\star,t-1} = \phi(g_{Q',\phi^{-1}(P)}^{\star,t-1}) = a' \cdot (\phi \circ \pi')(g_{Q',\phi^{-1}(P)}) = a' \cdot \pi(g_{Q,P}),$$

hence

$$a = a' = \frac{g_{[t-1]Q',\phi^{-1}(P)}}{\pi'(g_{Q',\phi^{-1}(P)})}.$$

Given that $Z_{\phi^{-1}(P)} = Z_{Q'} = Z_{\phi^{-1}(P)+Q'} = 1$ and $X_{\phi^{-1}(\mathcal{O}_E)} = 1/\xi^2$, the reduced ate pairing on $K$ can be computed as

$$\begin{aligned} f_{t-1,Q}(P)^{\frac{2(p^k-1)}{r}} = a'^{\frac{p^k-1}{r}} &= \left(\frac{g_{[t-1]Q',\phi^{-1}(P)}}{\pi'(g_{Q',\phi^{-1}(P)})}\right)^{\frac{p^k-1}{r}} \\ &= \left(\frac{Z_{[t-1]Q'+\phi^{-1}(P)}}{Z_{[t-1]Q'} \cdot \xi^2}\right)^{\frac{p^k-1}{r}}. \end{aligned}$$

Since $Z_{[t-1]Q'}$ and $\xi^2$ lie in the subfield $\mathbb{F}_{p^{k/d}}$, they vanish in the final exponentiation. Thus, it suffices to compute

$$a_b(P, Q) = a_\lambda(P, Q)^2 = Z_{[t-1]Q'+\phi^{-1}(P)}^{\frac{p^k-1}{r}},$$

which completes the proof. □

By utilizing the twisting isomorphism, part of the computations can be performed in the smaller field $\mathbb{F}_{p^{k/d}}$. The coordinate $Z_{[t-1]Q'+\phi^{-1}(P)}$ 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 several curves the ate pairing itself is optimal. We provide the following example for illustration.

**Example 2** (BLS12 Family)**.** *The BLS12 family, with an 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 $\phi : E' \to E$, $(x, y) \mapsto (D^{\frac{1}{3}}x, D^{\frac{1}{2}}y)$ with $D \in \mathbb{F}_{p^2}^* \mod (\mathbb{F}_{p^2}^*)^6$. By Theorem 1, the ate pairing on $E/\langle\pm1\rangle$ via biextension can be computed as*

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

### 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 $|c_i| \approx r^{\frac{1}{\varphi(k)}}$. According to [?, Section 3.4], the formula for the optimal ate pairing in Eq. (3) on $\mathbb{G}_2 \times \mathbb{G}_1$ through biextension can be expressed as

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

Similar to the ate pairing, the technique of twists can also be employed to enhance computational efficiency. We now present the following theorem to illustrate the formulas for optimal ate pairings on Kummer lines through biextensions 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 $K = E/\langle\pm1\rangle$ corresponding to the biextension $X_{2(\mathcal{O}_E)}$ can be computed as*

$$opt_b(P, Q) = opt(P, Q)^2 = \left(\prod_{\star_1, i=0}^{l} \pi^i(g_{([c_i]Q'),P})\right)^{\frac{p^k-1}{r}}$$

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

*Proof.* For any $i, j \in \mathbb{Z},\ i \neq j$, the following two equations hold

$$g_{[c_ip^i]Q,P} \star_1 g_{[c_jp^j]Q,P} = g_{[c_ip^i]Q,P}(\cdot) g_{[c_jp^j]Q,P}(\cdot) \frac{g_{[c_ip^i]Q,[c_jp^j]Q}(P+\cdot)}{g_{[c_ip^i]Q,[c_jp^j]Q}(\cdot)}$$
$$g^{p^i}_{[c_i]Q,P} \star_1 g^{p^j}_{[c_j]Q,P} = g^{p^i}_{[c_i]Q,P}(\cdot) g^{p^j}_{[c_j]Q,P}(\cdot) \frac{g_{[c_i]\pi^i(Q),[c_j]\pi^j(Q)}(P+\cdot)}{g_{[c_i]\pi^i(Q),[c_j]\pi^j(Q)}(\cdot)},$$

which implies that

$$g_{[c_ip^i]Q,P} \star_1 g_{[c_jp^j]Q,P} = \left(g^{p^i}_{[c_i]Q,P} \star_1 g^{p^j}_{[c_j]Q,P}\right) \cdot a_i \cdot a_j,$$

where $a_i = \frac{g_{[c_ip^i]Q,P}(\cdot)}{g^{p^i}_{[c_i]Q,P}(\cdot)},\ a_j = \frac{g_{[c_jp^j]Q,P}(\cdot)}{g^{p^j}_{[c_j]Q,P}(\cdot)}$ give two ate$_i$ pairings [**?**]. By induction, we deduce that the monodromy of the optimal pairing is

$$\prod_{\star_1, i=0}^{l} g_{[c_ip^i]Q,P} = \prod_{\star_1, i=0}^{l} g^{p^i}_{[c_i]Q,P}(\cdot) \cdot \left(\prod_{i=0}^{l} a_i\right) = C \cdot \left(\prod_{i=0}^{l} a_i\right),$$

where $a_i$ are a series of ate$_i$ pairings, and $C$ corresponds to the optimal ate pairing. Now we consider the twist $E'$, the corresponding biextension functions on $E'$ are given as follows

$$\phi^*(g_{[c_ip^i]Q,P}) = g_{[c_ip^i]Q',\phi^{-1}(P)},\ \phi^*\left(g^{p^i}_{[c_i]Q,P}\right) = g^{p^i}_{[c_i]Q',\phi^{-1}(P)}.$$

Thus the monodromy on the biextension $\mathrm{X}_{2(\mathcal{O}_{E'})}$ is

$$\prod_{\star_1, i=0}^{l} g_{[c_ip^i]Q',\phi^{-1}(P)} = \prod_{\star_1, i=0}^{l} g^{p^i}_{[c_i]Q',\phi^{-1}(P)}(\cdot) \cdot \left(\prod_{i=0}^{l} a'_i\right) = C' \cdot \left(\prod_{i=0}^{l} a'_i\right).$$

Since $\prod_{\star_1, i=0}^{l} g_{[c_ip^i]Q',\phi^{-1}(P)} = g_{[mr]Q',\phi^{-1}(P)}$ gives a Tate pairing, it follows from [**?**, Remark 3.22] that

$$\prod_{\star_1, i=0}^{l} g_{[c_ip^i]Q',\phi^{-1}(P)} = \prod_{\star_1, i=0}^{l} g_{[c_ip^i]Q,P}.$$

Besides, by Theorem 1 we obtain $a'_i = a_i$. We deduce that

$$\prod_{\star_1, i=0}^{l} g^{p^i}_{[c_i]Q,P}(\cdot) = \prod_{\star_1, i=0}^{l} g^{p^i}_{[c_i]Q',\phi^{-1}(P)}(\cdot) = \prod_{\star_1, i=0}^{l} \pi^i(g_{[c_i]Q',\phi^{-1}(P)}(\cdot)),$$

which completes the proof. □

From the above proof, we require the computation of the following coordinates

$$Z_{\pi^i([c_i]Q'+\phi^{-1}(P))} \quad \text{and} \quad Z_{\pi^i([c_i]Q')}, \quad i = 0, \dots, l,$$

which can be achieved through the following steps

1. Compute $Z_{[c_i]Q'}$ and $Z_{[c_i]Q'+\phi^{-1}(P)}$ for $i = 0, \dots, l$ using the cubical ladder algorithm.
2. Apply the morphisms $\pi^i$ separately to the points $[c_i]Q'$ and $[c_i]Q' + \phi^{-1}(P)$ to obtain $Z_{\pi^i([c_i]Q')}$ and $Z_{\pi^i([c_i]Q'+\phi^{-1}(P))}$.
3. Compute $Z_{\sum_{i=0}^{l} \pi^i([c_i]Q')}$ and $Z_{\sum_{i=0}^{l} \pi^i([c_i]Q'+\phi^{-1}(P))}$ from the points $Z_{\pi^i([c_i]Q')}$ and $Z_{\pi^i([c_i]Q'+\phi^{-1}(P))}$ using the three-way addition algorithm [?].

The most computationally expensive step is the calculation of $Z_{[c_i]Q'}$ and $Z_{[c_i]Q'+\phi^{-1}(P)}$. By employing the technique of twists, part of the computation can be performed over the smaller field $\mathbb{F}_{p^{k/d}}$, compared to the original approach in [?]. Detailed algorithms and cost analysis are provided in Section 4.2. In the following, we present the AFG16 family as a concrete example.

**Example 3** (AFG16 Family)**.** *The AFG16 family, with an 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 $D \in \mathbb{F}_{p^4}^*$ $\bmod (\mathbb{F}_{p^4}^*)^4$. Additionally, it holds that $z + p^5 \equiv 0 \mod r$ on the AFG16 family. By Theorem 2, the optimal ate pairing on AFG16 can be computed through biextension as:*

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

*In practice, this formula can be further simplified, and the corresponding details are provided in Section 4.2.*

### 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-endomorphisms. 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.

#### 3.4.1 The Super-optimal Ate Pairing on Curves Admitting Twists

Using the notation as above, let $\phi$ denote a degree-$d$ twist. 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:

$$E_1 : y^2 = x^3 + b, \quad j(E_1) = 0,$$
$$E_2 : y^2 = x^3 + ax, \quad j(E_2) = 1728.$$

Assume that the embedding degree $k$ satisfies $\gcd(k, d) \neq 1$. Let $\Phi = \phi^{-1} \circ \sigma \circ \phi$ and $\psi = \phi^{-1} \circ \pi \circ \phi$ denote the two endomorphisms on $E'_i$ $(i = 1, 2)$, where $\sigma$ and $\pi$ represent the efficiently-computable automorphism and Frobenius map on $E_i$ $(i = 1, 2)$, respectively. Recalled from Section 2, the pairing subgroup $\mathbb{G}_2$ in $E_i$ $(i = 1, 2)$ can be represented as

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

According to [?, ?], $E_1$ (resp. $E_2$) precisely to a pairing-friendly curve in the Cyclo (6.6) family (resp. Cyclo (6.3), (6.4), or (6.5)) parametrized by $(p(z), t(z), r(z))$ in [?]. Additionally, as noted in [?], for any $Q \in \mathbb{G}_2 \subseteq E_i(\mathbb{F}_{p^k})$ and $Q = \phi(Q')$, there exists a positive integer $j$ $(1 \leq j < k)$ such that

$$\psi^j \circ \Phi(Q') = [z]Q'.$$

If the embedding degree $k$ satisfies $\mathrm{ord}(\sigma) \nmid k$, it enables us to reduce the number of Miller iterations to approximately $\log_2(r)/2\varphi(k)$ and construct the corresponding super-optimal ate pairings. Our results are presented in the following theorem.

**Theorem 3.** *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 $\sigma$ 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*

$$\psi^j \circ \Phi(Q') = [z]Q'$$

*for any $Q' \in E'_i(\mathbb{F}_{p^{k/d}})[r]$. Assume that $P \in E_i(\mathbb{F}_p)[r]$. Then, the super-optimal ate pairing on $K_i = E_i/\langle\pm1\rangle$ $(i = 1, 2)$ corresponding to the biextension $X_{2(\mathcal{O}_E)}$ with sections $(X, Z)$ can be executed as follows*

1. *If we are on $K_1 = E_1/\langle\pm1\rangle$ with $j(E_1) = 0$, then the formula for the super-optimal ate pairing is*

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

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*

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

*Proof.* For simplicity, we prove the first case. The formulas for the super-optimal ate pairing on $E_2$ can be derived similarly. Define $\tau$ as the composition of the maps $\psi^j$ and $\Phi$ on $E'_i$ $(i = 1, 2)$. Let $\zeta_k$ denote the $k$-th roots of unity. By the characteristic equations of $\Phi = \phi^{-1} \circ \sigma \circ \phi$ and $\psi = \phi^{-1} \circ \pi \circ \phi$, we observe that $\Phi$ and $\psi$ correspond to $\frac{-1\pm\sqrt{-3}}{2}$ and $\zeta_k$ in $\mathrm{End}(E'_1)$, respectively. Consequently, we obtain

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

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

$$z^2 + zp^j + p^{2j} \equiv 0 \mod r. \tag{8}$$

By Eq. (3) and $\phi(Q') = Q$, we can derive a super-optimal ate pairing

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

To construct the formulas utilizing biextension, we require the following lemma to establish the relationship between the rational function $f_{n,P}$ evaluated on the divisor $(\cdot + Q) - (\cdot)$ and $g_{[n]P,Q}(\cdot)$ for any $n \in \mathbb{N}_+$.

**Lemma 2.** *Let the notations be as above. For any positive integer $n$ and $(P,Q) \in E(\mathbb{F}_{p^k})[r] \times E(\mathbb{F}_{p^k})/rE(\mathbb{F}_{p^k})$, it holds that*

$$\frac{g_{[n]P,Q}(\cdot)}{g^n_{P,Q}(\cdot)} = f^2_{n,P}((\cdot + Q) - (\cdot)) \cdot \frac{v^2_{[n]P}((\cdot + Q) - (\cdot))}{v^{2n}_P((\cdot + Q) - (\cdot))}, \tag{9}$$

*where $v_P$ is the vertical line passing through $P$ and $-P$.*

*Proof.* According to [?, Eq. (14)], in level 1 it follows that

$$g^{\star 1,n}_{P,Q} = g_{[n]P,Q}(\cdot) = g^n_{P,Q}(\cdot) \cdot f_{n,P}((Q+\cdot)-(\cdot)).$$

Going to level 2, it can be rewritten as

$$\frac{g_{[n]P,Q}(\cdot)}{g^n_{P,Q}(\cdot)} = \frac{f^2_{n,-P}(\cdot)}{f^2_{n,-P}(\cdot + Q)}.$$

Furthermore, it can be deduced that

$$\begin{aligned} \operatorname{div}\left(f_{n,P}(\cdot)^2 \cdot f_{n,-P}(\cdot)^2\right) &= 2n((P)+(-P)-2(\mathcal{O}_E)) - 2(([n]P)+([-n]P)-2(\mathcal{O}_E)) \\ &= \operatorname{div}\left(\frac{v^{2n}_P(\cdot)}{v^2_{[n]P}(\cdot)}\right). \end{aligned}$$

Thus it yields that

$$f_{n,P}(\cdot)^2 \cdot f_{n,-P}(\cdot)^2 = \lambda \cdot \frac{v^{2n}_P(\cdot)}{v^2_{[n]P}(\cdot)},\ \lambda \in \mathbb{F}_{p^k} \text{ is a constant.}$$

The constant $\lambda$ vanishes while evaluating on the divisor $(Q+\cdot)-(\cdot)$

$$\begin{aligned} f_{n,P}((Q+\cdot)-(\cdot))^2 \cdot f_{n,-P}((Q+\cdot)-(\cdot))^2 &= \frac{\lambda \cdot \frac{v^{2n}_P((Q+\cdot))}{v^2_{[n]P}((Q+\cdot))}}{\lambda \cdot \frac{v^{2n}_P((\cdot))}{v^2_{[n]P}((\cdot))}} \\ &= \frac{v^{2n}_P((Q+\cdot)-(\cdot))}{v^2_{[n]P}((Q+\cdot)-(\cdot))}. \end{aligned}$$

Consequently, we obtain

$$\begin{aligned} \frac{g_{[n]P,Q}(\cdot)}{g^n_{P,Q}(\cdot)} &= \frac{f^2_{n,-P}(\cdot)}{f^2_{n,-P}(\cdot + Q)} \\ &= f^2_{n,P}((\cdot + Q)-(\cdot)) \cdot \frac{v^2_{[n]P}((\cdot + Q)-(\cdot))}{v^{2n}_P((\cdot + Q)-(\cdot))}, \end{aligned}$$

which completes the proof. □

It is worth noting that Lemma 2 can be regarded as the generalization of Lemma 1 for any $n \in \mathbb{N}_+$ besides $n = r$. Additionally, Eq. (9) can be simply written as

$$\frac{g_{[n]P,Q}}{g^n_{P,Q}} = f^2_{n,P}(Q) \cdot \frac{v^2_{[n]P}(Q)}{v^{2n}_P(Q)}$$

while evaluating on the point $Q$. Since we are working with the curve admitting with a twist, the evaluation of the vertical function $v^2_{[n]P}/v^{2n}_P$ lies in $\mathbb{F}_{p^{k/d}}$. Thus it vanishes while performing the final exponentiation. We suffice to compute

$$\frac{g_{[n]P,Q}}{g^n_{P,Q}} = f^2_{n,P}(Q). \tag{10}$$

Additionally, according to [?, Section 3.3], it holds that

$$\mathrm{div}(g_{P,Q}(\cdot)) = \mathrm{div}\left(\frac{v^2_{P+Q}(\cdot)}{\ell^2_{-P,-Q}(\cdot)}\right), \tag{11}$$

which implies that $g_{P,Q}(\cdot) = \lambda \cdot \frac{v^2_{P+Q}(\cdot)}{\ell^2_{-P,-Q}(\cdot)}$, where $\lambda$ is a constant. Therefore, Eq. (4) can be rewritten as

$$\begin{aligned}
g_{-P,Q}(\cdot) &= \frac{1}{g_{P,Q}(\cdot)} \cdot \frac{g_{-P,P}(\cdot)}{g_{-P,P}(\cdot + Q)} \\
&= \frac{1}{g_{P,Q}(\cdot)} \cdot \frac{\lambda \cdot \frac{1}{v^2_P(\cdot)}}{\lambda \cdot \frac{1}{v^2_P(Q+\cdot)}} \\
&= \frac{1}{g_{P,Q}(\cdot)} \cdot \frac{v^2_P(\cdot + Q)}{v^2_P(\cdot)}.
\end{aligned}$$

Similarly, we can obtain

$$g_{P,-Q} = \frac{1}{g_{P,Q}} \cdot \frac{v^2_Q(\cdot + P)}{v^2_Q(\cdot)}.$$

As the evaluations of the vertical functions vanish in the process of the final exponentiation, we deduce that

$$(g_{P,Q})^{\frac{p^k-1}{r}} = \left(\frac{1}{\ell^2_{-P,-Q}(\cdot)}\right)^{\frac{p^k-1}{r}},$$

$$(g_{-P,-Q})^{\frac{p^k-1}{r}} = \left(\frac{1}{g_{P,-Q}} \cdot \frac{v^2_P(\cdot - Q)}{v^2_P(\cdot)}\right)^{\frac{p^k-1}{r}} = \left(g_{P,Q} \cdot \frac{v^2_Q(\cdot)}{v^2_Q(\cdot + P)}\right)^{\frac{p^k-1}{r}} = (g_{P,Q})^{\frac{p^k-1}{r}}$$

By Eqs. (8), (10), (11) and the definition of $g_{P,Q}$, the square of the above super-optimal pairing can be derived as

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

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

$$= \left( \left( \frac{g_{[z]Q,P}}{g^{p^j}_{Q,\sigma^{-1}(P)}} \right)^z \cdot \left( g_{[z]Q,P} \cdot g_{[z]Q,\sigma^{-1}(P)} \cdot g^{p^j}_{Q,P} \right)^{p^j} \cdot \ell^2_{[p^{2j}]Q,[zp^j]Q}(P) \right)^{\frac{p^k-1}{r}}$$

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

Due to the fact that $Z_{[z]Q} \in \mathbb{F}_{p^{k/d}}$ and it vanishes during the final exponentiation, the above formula for the super-optimal ate pairing through biextension can be simplified as

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

It remains to exploit the twisting isomorphism to further enhance the efficiency of this pairing. By Theorem 2, the super-optimal ate pairing can be computed as follows

$$\left( \frac{g_{[z]Q,P}}{g^{p^j}_{Q,\sigma^{-1}(P)}} \right)^z \cdot \left( g_{[z]Q,P} \cdot g_{[z]Q,\sigma^{-1}(P)} \cdot g^{p^j}_{Q,P} \right)^{p^j} \cdot \ell^2_{[p^{2j}]Q,[zp^j]Q}(P)$$

$$= \left( \frac{g_{[z]Q',\phi^{-1}(P)}}{g^{p^j}_{Q',\sigma^{-1}\circ\phi^{-1}(P)}} \right)^z \cdot \left( g_{[z]Q',\phi^{-1}(P)} \cdot g_{[z]Q',\sigma^{-1}\circ\phi^{-1}(P)} \cdot g^{p^j}_{Q',\phi^{-1}(P)} \right)^{p^j} \cdot \ell^2_{[p^{2j}]Q',[zp^j]Q'}(\phi^{-1}(P))$$

$$= \left( \frac{Z^{z+2p^j}_{[z]Q'+\phi^{-1}(P)} \cdot Z^{p^j}_{[z]Q'+\sigma^{-1}\circ\phi^{-1}(P)}}{Z^{p^j}_{\pi^j(Q)+[z]Q+P}} \right),$$

which completes the proof. □

From the above theorem, the most costly part of computing the super-optimal pairing through biextension is to obtain the two coordinates $Z_{[z]Q'+\phi^{-1}(P)}$ and $Z_{[z]Q'+\sigma^{-1}\circ\phi^{-1}(P)}$. This can be done by utilizing the cubical ladder [?]. It is worth noting that the calculation of the coordinate $Z_{[z]Q'}$ is over the subfield $\mathbb{F}_{p^{k/d}}$ and can be shared during the computations of $Z_{[z]Q'+\phi^{-1}(P)}$ and $Z_{[z]Q'+\sigma^{-1}\circ\phi^{-1}(P)}$. The detailed computational procedure is presented in Section 4.3. In the following we provide an example for illustration.

**Example 4** (BW14 family)**.** *As mentioned in [?], 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*

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

*There exists a quadratic twist $E'$ for a BW14 curve $E$. From Section 2.3, the twisting isomorphism is defined as $\phi : E' \to E,\ (x, y) \mapsto (Dx, D^{\frac{3}{2}}y)$ 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 3 the super-optimal ate pairing on BW14 can be obtained through biextension as*

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

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

For the pairing-friendly curves with the lack of twists, 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 $\mathbb{F}_{p^k}$. We also consider the pairing-friendly curves $E_1$ and $E_2$.

Based on the above analysis, the embedding degree $k$ must be odd. The curve $E_1$ (resp. $E_2$) corresponds to a pairing-friendly curve in Cyclo (6.6) (resp. Cyclo (6.3) or (6.4) or (6.5)), which is parametrized by $(p(z), t(z), r(z))$ [**?**]. Similarly, for $Q \in \mathbb{G}_2 \subseteq E_i(\mathbb{F}_{p^k})$ there is a positive integer $j$ $(1 \le j < k)$ 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 is presented in Theorem 4.

**Theorem 4.** *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 $\sigma$ such that $\mathrm{ord}(\sigma) \nmid k$. Let $E_i$ $(i = 1, 2)$ be a curve in this family such that there exist a positive integer $j$ $(1 \le j < k)$ satisfying*

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

*for any $Q \in E_i(\mathbb{F}_{p^k})[r] \cap \ker(\pi - [p])$. Assume that $P \in E_i(\mathbb{F}_p)[r]$. 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*

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

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*

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

*Proof.* We only provide the proof of the formula on $K_1 = E_1/\langle \pm 1\rangle$ for simplicity. By Eqs. (3) and (8), the super-optimal pairing on $E_1$ is

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

Furthermore, from Lemma 2 we know that

$$\operatorname{div}\left(f^2_{z,Q}(P)\right) = \operatorname{div}\left(\frac{g^z_{-Q,P}}{g_{[-z]Q,P}}\right).$$

By the definition of the function $g_{Q,P}$, the square of the pairing can be represented as

$$\begin{aligned} & sopt(Q,P)^2 \\ = & \left( \frac{g^{z^2+zp^j}_{-Q,P}}{g^{z+p^j}_{[-z]Q,P}} \cdot \frac{g^{zp^j}_{-Q,\sigma^{-1}(P)}}{g^{p^j}_{[-z]Q,\sigma^{-1}(P)}} \cdot \ell^2_{[p^{2j}]Q,[zp^j]Q}(P) \right)^{\frac{p^k-1}{r}} \\ = & \left( \frac{\ell^2_{[p^{2j}]Q,[zp^j]Q}(P)}{g^{z+p^j}_{[-z]Q,P} \cdot g^{p^j}_{[-z]Q,\sigma^{-1}(P)}} \right)^{\frac{p^k-1}{r}} \\ = & \left( \frac{Z^{z+2p^j}_{[-z]Q} \cdot \ell^2_{[p^{2j}]Q,[zp^j]Q}(P)}{Z^{z+p^j}_{[-z]Q+P} \cdot Z^{p^j}_{[-z]Q+\sigma^{-1}(P)}} \right)^{\frac{p^k-1}{r}} \end{aligned}$$

which completes the whole proof. □

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 ladder. Now we present the following family BW13 for description.

**Example 5** (BW13 family)**.** *From [?], 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 [?] 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 4 the super-optimal ate pairing on the above family can be obtained through biextension as*

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

*As for $D = 3$, the following three polynomials parameterize a family of pairing-friendly curves with embedding degree $k = 13$* (*see Cyclo* (6.6) *[?] 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 $\pi \circ \sigma(Q) = [z]Q$. From Theorem 4, the formula of the super-optimal on this family is*

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

# 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 implement algorithms to the approaches in [?] and the classical Miller's algorithm. According to the analysis in Section 3, we give 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 $z$ and the map $\phi$ are the parametrized seed and the twisting isomorphism of the family of the pairing-friendly curves, respectively.

| $k$ | Curve | Pairing formula through biextension |
|---|---|---|
| 12 | BLS12, $D = 3$ | $Z_{[z]Q'+\phi^{-1}(P)}^{\frac{p^{12}-1}{r}}$ |
| 16 | AFG16, $D = 1$ | $\left(\frac{Z_{[z]Q'+\pi^5(Q')+\phi^{-1}(P)}}{Z_{[z]Q'+\pi^5(Q')}}\right)^{\frac{p^{16}-1}{r}}$ |
| 14 | BW14, $D = 1$ | $\left(Z_{[z]Q'+\phi^{-1}(P)}^{z} \cdot Z_{[z]Q'+\sigma^{-1}\circ\phi^{-1}(P)}^{p^4}\right)^{\frac{p^{14}-1}{r}}$ |
| 14 | BW14, $D = 3$ | $\left(\frac{Z_{[z]Q'+\phi^{-1}(P)}^{z+2p} \cdot Z_{[z]Q'+\sigma^{-1}\circ\phi^{-1}(P)}^{p}}{Z_{\pi(Q)+[z]Q+P}^{p}}\right)^{\frac{p^{14}-1}{r}}$ |
| 13 | BW13, $D = 1$ | $\left(\frac{Z_{[-z]Q}^{z+p^7} \cdot v_Q(P)^{2p}}{Z_{[-z]Q+P}^{z} \cdot Z_{[-z]Q+\sigma^{-1}(P)}^{p^7}}\right)^{\frac{p^{13}-1}{r}}$ |
| 13 | BW13, $D = 3$ | $\left(\frac{Z_{[-z]Q}^{z+2p} \cdot \ell_{[p^2]Q,[zp]Q}^{2}(P)}{Z_{[-z]Q+P}^{z+p} \cdot Z_{[-z]Q+\sigma^{-1}(P)}^{p}}\right)^{\frac{p^{13}-1}{r}}$ |

In the following, we provide the detailed computational procedure and cost analysis for the formulas above.

**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 calculation of the additions 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 for the Miller's function evaluation of the ate pairing on BLS12 family with $D = 3$. 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 suffices to compute

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

where $\phi$ 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 3.2, we can employ the cubical Montgomery ladder (See [**?**, Algorithm 4.2]) to derive $Z_{[z]Q'+\phi^{-1}(P)}$. Additionally, the double-and-add ladder can also be utilized to compute this coordinate. Compared to the cubical ladder, the double-and-add ladder allows for one less differential addition calculation in each doubling step, with an expensive compatible addition in the double-and-add step. Since the Hamming weight of the parametrized seed $z$ is relatively small, we prefer the double-and-add ladder for efficiency purpose.

Denote by $\texttt{xDBL}(P)$, $\texttt{xDIFF}(P, Q, P - Q)$ and $\texttt{xADD}(P_1, P_2, P_1 + Q, P_2 - Q)$ the algorithms of $x$-only point doubling, differential addition and compatible addition (See Algorithms 5, 6 and 9 for more details) on the Kummer line $K = E/\langle \pm 1 \rangle$ with $j(E) = 0$, respectively. The detailed computational procedure are given in Algorithm 1.

**Algorithm 1** The double-and-add ladder to compute $Z_{[n]Q'+\phi^{-1}(P)}$

**Input:** The points $Q' = (X_{Q'} : Z_{Q'})$, $\phi^{-1}(P) = (X_{\phi^{-1}(P)} : Z_{\phi^{-1}(P)})$, $Q' + \phi^{-1}(P) = (X_{Q'+\phi^{-1}(P)} : Z_{Q'+\phi^{-1}(P)}) \in E'$. The inverses of the $X$-coordinates of $Q', \phi^{-1}(P)$ and $Q' - \phi^{-1}(P)$: $iX_{Q'},\ iX_{\phi^{-1}(P)},\ iX_{Q'-\phi^{-1}(P)}$. The scalar $n$ ($n > 2$) that needs to be performed. Assume that $n = \sum_{i=0}^{N} n_i 2^i$.

**Output:** The point $[n]Q' + \phi^{-1}(P) = (X_{[n]Q'+\phi^{-1}(P)} : Z_{[n]Q'+\phi^{-1}(P)})$

```
1:  $R \leftarrow Q',\ S \leftarrow Q' + \phi^{-1}(P)$
2:  for $i = N - 1$ to 0 do                          ▷ $R = [k]Q',\ S = [k]Q' + \phi^{-1}(P)$
3:      if $n_i = 0$ then
4:          $R \leftarrow \texttt{xDBL}(R)$
5:          $S \leftarrow \texttt{xDIFF}(S, R, iX_{\phi^{-1}(P)})$
6:      else
7:          $T \leftarrow \texttt{xADD}(R, Q', S, Q' - \phi^{-1}(P))$          ▷ $T = [k+1]Q'$
8:          $R \leftarrow \texttt{xDIFF}(T, R, iX_{Q'})$
9:          $S \leftarrow \texttt{xDIFF}(T, S, iX_{Q'-\phi^{-1}(P)})$
10:     end if
11: end for
12: return $S$
```

Now we analyze the cost for each basic iteration step during the computation of $Z_{[z]Q'+\phi^{-1}(P)}$. According to Algorithm 1, we know that it takes a point doubling and a differential addition to execute in the doubling step (the bit is 0). More precisely, the point doubling (Line 4 in Algorithm 1) are performed in $E'(\mathbb{F}_{p^2})$, while the differential addition (Line 5 in Algorithm 1) is executed over the whole extension field $\mathbb{F}_{p^{12}}$.

Note that in the phase of the $\texttt{xDBL}$ over $\mathbb{F}_{p^2}$, the coefficient of $E'$ is $b' = b/u$. If $b$ is small, multiplying an element in $\mathbb{F}_{p^2}$ by $b'$ can be roughly regarded as a shifting operation, whose cost is negligible. According to Algorithm 5, the corresponding cost is

$$\mathrm{Costdbl} = 4\mathbf{m}_2 + 2\mathbf{s}_2.$$

Additionally, if the bit is 0, the multiplication by $1/X_{\phi^{-1}(P)}$ can also be regarded as a shifting operation in `xDIFF` 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 6 the cost of Line 5 in Algorithm 1 is

$$\mathrm{Costdiff}_{\mathrm{bit0}} = \mathbf{m}_{12} + 2\mathbf{s}_{12} + 4 \cdot 6\mathbf{m}_2.$$

Hence, the cost for a doubling step is

$$\mathrm{Cost}_{\mathrm{bit0}} = \mathrm{Costdbl} + \mathrm{Costdiff}_{\mathrm{bit0}} = \mathbf{m}_{12} + 2\mathbf{s}_{12} + 28\mathbf{m}_2 + 2\mathbf{s}_2.$$

As for the double-and-add step (the bit is 1), it requires one compatible addition, and two differential additions. One of the differential addition (Line 8 in Algorithm 1) is executed over $\mathbb{F}_{p^2}$, while the other (Line 9 in Algorithm 1) is over $\mathbb{F}_{p^{12}}$. The corresponding costs are

$$\mathrm{Costdiff}_{\mathbb{F}_{p^2}} = 6\mathbf{m}_2 + 2\mathbf{s}_2, \ \mathrm{Costdiff}_{\mathrm{bit1}} = 2\mathbf{m}_{12} + 2\mathbf{s}_{12} + 4 \cdot 6\mathbf{m}_2.$$

It is worth noting that part of the computation of the compatible addition is carried out over the extension field $\mathbb{F}_{p^{12}}$. From Algorithm 9 we can calculate the cost as follows

$$\mathrm{Costadd} = 4\mathbf{m}_{12} + 3\mathbf{s}_{12} + 28\mathbf{m}_2 + 3\mathbf{s}_2.$$

Based on the above analysis, the computational cost for a double-and-add iteration step is

$$\mathrm{Cost}_{\mathrm{bit1}} = \mathrm{Costdiff}_{\mathbb{F}_{p^2}} + \mathrm{Costdiff}_{\mathrm{bit1}} + \mathrm{Costadd} = 6\mathbf{m}_{12} + 5\mathbf{s}_{12} + 58\mathbf{m}_2 + 5\mathbf{s}_2.$$

**Remark 1.** *The compatible addition makes the coordinates of $(X_T : Z_T)$ lie in the full extension field $\mathbb{F}_{p^{12}}$. We consider constructing a linear form $f : \mathbb{F}_{p^{12}} \to \mathbb{F}_{p^2}$ (Note that $\mathbb{F}_{p^{12}}$ can be regarded as a linear space of $\mathbb{F}_{p^2}$) that fixes the elements in $\mathbb{F}_{p^2}$. By acting $f$ on $(X_T : Z_T)$, we have*

$$(f(X_T) : f(Z_T)) = (\frac{X_T}{Z_T} \cdot f(Z_T) : f(Z_T)),$$

*which is over $\mathbb{F}_{p^2}$ since $x_T = \frac{X_T}{Z_T} \in \mathbb{F}_{p^2}$. Through this adjustment, the subsequent operations can be performed within the subfield $\mathbb{F}_{p^2}$.*

## 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 $\alpha \in \mathbb{F}_p$. From Table 1 we can see that it suffices to compute $Z_{[z]Q'+\phi^{-1}(P)}$, where $\phi$ is a degree-4 twist isomorphism

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

The curve $E$ and its twist $E'$ are $y^2 = x^3+ax$ and $y^2 = x^3 + a/u \cdot x$, respectively. Recalled from Table 1 and Example 3, we can deduce that $z^2+p^5 \equiv 0 \mod r$, and the optimal pairing on AFG16 family through biextension can be derived as

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

In fact, the above formula can be further simplified. By Eq. (3), the optimal pairing on this curve can be obtained as $opt(P,Q) = f_{z,Q}(P)^{\frac{p^{16}-1}{r}}$. Consequently, by utilizing Eq. (10) it yields that

$$opt_b(P,Q) = \left(\frac{g_{[z]Q,P}}{g^z_{Q,P}}\right)^{\frac{p^{16}-1}{r}} = \left(\frac{g_{[z]Q,P}\cdot g^{p^5}_{Q,P}}{g^{z+p^5}_{Q,P}}\right)^{\frac{p^{16}-1}{r}} = Z^{\frac{p^{16}-1}{r}}_{[z]Q'+\phi^{-1}(P)}.$$

Hence, it requires to compute $Z_{[z]Q'+\phi^{-1}(P)}$, which can also be done by exploiting Algorithm 1. Additionally, the cost for multiplying two elements in $\mathbb{F}_{p^4}$ and $\mathbb{F}_{p^{16}}$ can be estimated as $4\mathbf{m}_4$. In the doubling step, it requires a point doubling over $\mathbb{F}_{p^4}$ and a differential addition over $\mathbb{F}_{p^{16}}$. From Algorithms 7 and 8, the costs for `xDBL` and `xDIFF` on Kummer line $K = E'/\langle\pm 1\rangle$ with $j(E') = 1728$ over $\mathbb{F}_{p^4}$ and $\mathbb{F}_{p^{16}}$ respectively are

$$\text{Costdbl} = 2\mathbf{m}_4 + 3\mathbf{s}_4,\ \text{Costdiff}_{\text{bit0}} = 2\mathbf{s}_{16} + 3\cdot 4\mathbf{m}_4.$$

Hence, the computational cost for a doubling step is

$$\text{Cost}_{\text{bit0}} = \text{Costdbl} + \text{Costdiff}_{\text{bit0}} = 2\mathbf{s}_{16} + 14\mathbf{m}_4 + 3\mathbf{s}_4.$$

As for the double-and-add step, we need to execute two differential additions (over $\mathbb{F}_{p^4}$ and $\mathbb{F}_{p^{16}}$) and a compatible addition. According to Algorithms 8 and 9 the corresponding costs are

$$\begin{aligned}&\text{Costdiff}_{\mathbb{F}_{p^4}} = 4\mathbf{m}_4 + 2\mathbf{s}_4,\ \text{Costdiff}_{\text{bit1}} = \mathbf{m}_{16} + 2\mathbf{s}_{16} + 3\cdot 4\mathbf{m}_4\\ &\text{Costadd} = 4\mathbf{m}_{16} + \mathbf{s}_{16} + 19\mathbf{m}_4 + \mathbf{s}_4\end{aligned}$$

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

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

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

We investigate the concrete computational processes for the super-optimal ate pairings on families BW14 and BW13 through biextension. Besides, we also present the computational cost analysis. 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

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

where $\phi$ is a degree-2 twist isomorphism

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

The curve $E$ and its twist $E'$ are $y^2 = x^3 + ax$ 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 obtain two coordinates $Z_{[z]Q'+\phi^{-1}(P)}$ and $Z_{[z]Q'+\sigma^{-1}\circ\phi^{-1}(P)}$. An intuitive approach is to separately compute them by Algorithm 1. Nevertheless, part of the computational modules can be shared. In the following, we state how to share the information as far as possible during the computations of $Z_{[z]Q'+\phi^{-1}(P)}$ and $Z_{[z]Q'+\sigma^{-1}\circ\phi^{-1}(P)}$ and provide optimized algorithms for description.

In each iteration of the double-and-add ladder in Algorithm 1, the $(X : Z)$-coordinates of $[k]Q'$ are both required in the phase of deriving $Z_{[z]Q'+\phi^{-1}(P)}$ and $Z_{[z]Q'+\sigma^{-1}\circ\phi^{-1}(P)}$. Consequently, we can accomplish the computation of these two coordinates in the same ladder, which is presented in Algorithm 2.

After calculating $Z_{[z]Q'+\phi^{-1}(P)}$ and $Z_{[z]Q'+\sigma^{-1}\circ\phi^{-1}(P)}$, we execute an exponentiation of $z$ and a Frobenius endomorphism over $\mathbb{F}_{p^{14}}$ to derive the final result $Z^{z}_{[z]Q'+\phi^{-1}(P)} \cdot Z^{p^4}_{[z]Q'+\sigma^{-1}\circ\phi^{-1}(P)}$. In fact, this exponentiation can be simultaneously

**Algorithm 2** The shared double-and-add ladder

```
Input: The points $Q' = (X_{Q'} : Z_{Q'})$, $\phi^{-1}(P) = (X_{\phi^{-1}(P)} : Z_{\phi^{-1}(P)})$, $Q' + \phi^{-1}(P) = (X_{Q'+\phi^{-1}(P)} : Z_{Q'+\phi^{-1}(P)})$ and $Q' + \sigma^{-1} \circ \phi^{-1}(P) = (X_{Q'+\sigma^{-1}\circ\phi^{-1}(P)} : Z_{Q'+\sigma^{-1}\circ\phi^{-1}(P)}) \in E'$. The inverses of the $X$-coordinates of $Q', \phi^{-1}(P)$, $Q' - \phi^{-1}(P)$ and $Q' - \sigma^{-1}\circ\phi^{-1}(P)$: $iX_{Q'}$, $iX_{\phi^{-1}(P)}$, $iX_{Q'-\phi^{-1}(P)}$ and $iX_{Q'-\sigma^{-1}\circ\phi^{-1}(P)}$.
    The scalar $n (n > 2)$ that needs to be performed. Assume that $n = \sum_{i=0}^{N} n_i 2^i$.
Output: The points $[n]Q' + \phi^{-1}(P) = (X_{[n]Q'+\phi^{-1}(P)} : Z_{[n]Q'+\phi^{-1}(P)})$ and $[n]Q' + \sigma^{-1} \circ \phi^{-1}(P) = (X_{[n]Q'+\sigma^{-1}\circ\phi^{-1}(P)} : Z_{[n]Q'+\sigma^{-1}\circ\phi^{-1}(P)})$.
$R \leftarrow Q'$, $S_1 \leftarrow Q' + \phi^{-1}(P)$, $S_2 \leftarrow Q' + \sigma^{-1} \circ \phi^{-1}(P)$
for $i = N - 1$ to $0$ do
    if $n_i = 0$ then        ▷ $R = [k]Q', S_1 = [k]Q' + \phi^{-1}(P), S_2 = [k]Q' + \sigma \circ \phi^{-1}(P)$
        $R \leftarrow$ xDBL($R$)
        $S_1 \leftarrow$ xDIFF($S_1, R, -iX_{\phi^{-1}(P)}$)
        $S_2 \leftarrow$ xDIFF($S_2, R, -iX_{\sigma^{-1}\circ\phi^{-1}(P)}$)
    else
        $T \leftarrow$ xADD($R, Q', S_1, Q' - \phi^{-1}(P)$)
        $R \leftarrow$ xDIFF($T, R, iX_{Q'}$)
        $S_1 \leftarrow$ xDIFF($T, S_1, iX_{Q'-\phi^{-1}(P)}$)
        $S_2 \leftarrow$ xDIFF($T, S_2, iX_{Q'-\sigma\circ\phi^{-1}(P)}$)
    end if
end for
return $S_1$, $S_2$
```

performed while executing the second ladder by utilizing the idea in [?, ?]. Now we describe how to generalize this method to our shared cubical ladder.

According to Algorithm 8, we can figure out the relationship between the points $P_1$, $P_2$ and $P_1 - P_2$:

$$\begin{aligned} X_{P_1+P_2} \cdot X_{P_1-P_2} &= (X_{P_1} \cdot X_{P_2} - aZ_{P_1} \cdot Z_{P_2})^2, \\ Z_{P_1+P_2} \cdot Z_{P_1-P_2} &= (X_{P_1} \cdot Z_{P_2} - X_{P_2} \cdot Z_{P_1})^2. \end{aligned}$$

On this basis, in a iteration we can update $X_{[k]Q'}$, $Z_{[k]Q'}$, $cX_{[k]Q'+\phi^{-1}(P)}$ and $cZ_{[k]Q'+\phi^{-1}(P)}$ as

$$\begin{aligned} &X_{[2k]Q'},\ Z_{[2k]Q'} \leftarrow \texttt{xDBL}([k]Q'), \\ &c^2 X_{[2k]Q'+\phi^{-1}(P)} = (X_{[k]Q'} \cdot cX_{[k]Q'+\phi^{-1}(P)} - aZ_{[k]Q'} \cdot cZ_{[k]Q'+\phi^{-1}(P)})^2 / X_{\phi^{-1}(P)}, \\ &c^2 Z_{[2k]Q'+\phi^{-1}(P)} = (X_{[k]Q'} \cdot cZ_{[k]Q'+\phi^{-1}(P)} - cX_{[k]Q'+\phi^{-1}(P)} \cdot Z_{[k]Q'})^2 \end{aligned}$$

if the bit is 0. The case where the bit is 1 follows a similar pattern. By acting a Frobenius endomorphism of power $p^{10}$ on the formula for the super-optimal ate pairing on BW14 family, we deduce that

$$\left(Z^{z}_{[z]Q'+\phi^{-1}(P)} \cdot Z^{p^4}_{[z]Q'+\sigma^{-1}\circ\phi^{-1}(P)}\right)^{\frac{p^{10}(p^{14}-1)}{r}} = \left(Z^{zp^{10}}_{[z]Q'+\phi^{-1}(P)} \cdot Z_{[z]Q'+\sigma^{-1}\circ\phi^{-1}(P)}\right)^{\frac{p^{14}-1}{r}}$$

also defines a bilinear pairing. Consequently, after finishing the first ladder to obtain $Z_{[z]Q'+\phi^{-1}(P)}$ we store the information that is also required for computing $Z_{[z]Q'+\sigma^{-1}\circ\phi^{-1}(P)}$. Moreover, we initialize $T_2 = (X_{T_2} : Z_{T_2})$ as $\left(Z^{2p^{10}}_{[z]Q'+\phi^{-1}(P)} \cdot X_{Q'+\sigma^{-1}\circ\phi^{-1}(P)} : Z^{2p^{10}}_{[z]Q'+\phi^{-1}(P)}\right)$ to execute the second ladder. The detailed procedure is presented in Algorithm 3. By employing this technique, we can save almost an exponentiation of $z$ during the computation of $Z^{zp^{10}}_{[z]Q'+\phi^{-1}(P)} \cdot Z_{[z]Q'+\sigma^{-1}\circ\phi^{-1}(P)}$.

**Algorithm 3** The optimized shared cubical ladder for computing $Z^{zp^{10}}_{[z]Q'+\phi^{-1}(P)} \cdot Z_{[z]Q'+\sigma^{-1}\circ\phi^{-1}(P)}$ on BW14 family

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

**Output:** The value $Z^{zp^{10}}_{[z]Q'+\phi^{-1}(P)} \cdot Z_{[z]Q'+\sigma^{-1}\circ\phi^{-1}(P)}$

```
$R \leftarrow Q'$, $S_1 \leftarrow Q' + \phi^{-1}(P)$
$tab_1 \leftarrow [\,]$, $j \leftarrow 0$
for $i = N - 1$ to $0$ do
    if $n_i = 0$ then
        $R \leftarrow \mathtt{xDBL}(R)$, $S_1 \leftarrow \mathtt{xDIFF}(S_1, R, iX_{\phi^{-1}(P)})$
        $tab_1[j] \leftarrow R$, $j \leftarrow j + 1$
    else
        $T \leftarrow \mathtt{xADD}(R, Q', S_1, Q' - \phi^{-1}(P))$, $R \leftarrow \mathtt{xDIFF}(T, R, iX_{Q'})$
        $S_1 \leftarrow \mathtt{xDIFF}(T, S_1, iX_{Q-\phi^{-1}(P)})$, $tab_1[j] \leftarrow T$, $j \leftarrow j + 1$
    end if
end for                                        ▷ $R = [z]Q'$, $S_1 = [z]Q' + \phi^{-1}(P)$
$j \leftarrow 0$, $c \leftarrow Z^{p^{10}}_{S_1}$, $S_2 \leftarrow (c \cdot X_{Q'+\sigma^{-1}\circ\phi^{-1}(P)} : c)$
for $i = N - 1$ to $0$ do
    if $n_i = 0$ then
        $S_2 \leftarrow \mathtt{xDIFF}(S_2, tab_1[j], -iX_{\phi^{-1}(P)})$, $j \leftarrow j + 1$
    else
        $S_2 \leftarrow \mathtt{xDIFF}(tab_1[j], S_2, c \cdot iX_{Q'-\sigma^{-1}\circ\phi^{-1}(P)})$, $j \leftarrow j + 1$
    end if
end for                                        ▷ $S_2 = [-z]Q' + \sigma^{-1} \circ \phi^{-1}(P)$
return $c \cdot Z_{S_2}$
```

Now we make a cost analysis for each iteration step. From Algorithm 3, it needs a point doubling over $\mathbb{F}_{p^7}$ and a two differential additions over $\mathbb{F}_{p^{14}}$ in the doubling step. From Algorithms 7 and 8, the corresponding costs are

$$\text{Costdbl} = 2\mathbf{m}_7 + 3\mathbf{s}_7, \ \text{Costdiff}_{\text{bit0}} = 2\mathbf{s}_{14} + 3 \cdot 2\mathbf{m}_7.$$

Thus the cost of a doubling step is

$$\mathrm{Cost}_{\mathrm{bit0}} = \mathrm{Costdbl} + 2\mathrm{Costdiff}_{\mathrm{bit0}} = 4\mathbf{s}_{14} + 14\mathbf{m}_7 + 3\mathbf{s}_7.$$

As for the double-and-add step, we need to execute three differential additions and a compatible addition. By Algorithms 8,9 and the analysis in Section 3.3, the corresponding costs are

$$\begin{aligned}&\mathrm{Costdiff}_{\mathbb{F}_{p^4}} = 4\mathbf{m}_7 + 2\mathbf{s}_7,\ \mathrm{Costdiff}_{\mathrm{bit1}} = \mathbf{m}_{14} + 2\mathbf{s}_{14} + 3 \cdot 2\mathbf{m}_7\\ &\mathrm{Costadd} = 4\mathbf{m}_{14} + \mathbf{s}_{14} + 11\mathbf{m}_7 + \mathbf{s}_7\end{aligned}$$

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

$$\mathrm{Cost}_{\mathrm{bit1}} = \mathrm{Costdiff}_{\mathbb{F}_{p^4}} + 2\mathrm{Costdiff}_{\mathrm{bit1}} + \mathrm{Costadd} = 6\mathbf{m}_{14} + 5\mathbf{s}_{14} + 27\mathbf{m}_4 + 3\mathbf{s}_7.$$

### 4.3.2 The super-optimal ate pairing on BW13 family

In this subsection, we provide the algorithm for the computation of super-optimal ate pairing on family BW13 with $D = 1$ exploiting biextension. The 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

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

where $\sigma$ is an automorphism

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

From [?], the double-and-add ladder can also be employed to compute $Z_{[-z]Q+P}$ and $Z_{[-z]Q+\sigma^{-1}(P)}$. During the whole ladder, we keep track on the following three values

$$Z_{[n]Q},\ Z_{[n]Q+P},\ Z_{[n]Q+\sigma^{-1}(P)}.$$

In the doubling step (bit $= 0$), we perform a point doubling and two differential additions to derive $[2n]Q,\ [2n]Q+P,\ [2n]Q+\sigma^{-1}(P)$ from $[n]Q,\ [n]Q+P$ and $[n]Q+\sigma^{-1}(P)$. While in the double-and-add step (bit $\neq 0$), we first execute a compatible addition (See Algorithm 9 for more details) to obtain $[n+1]Q$ from $[n]Q,\ P,\ [n]Q+P$ and $Q-P$, then we perform three differential additions to derive $[2n+1]Q,\ [2n+1]Q+P$ and $[2n+1]Q+\sigma^{-1}(P)$. Compared to the cubical ladder, this algorithm executes one less differential addition in each doubling step, but with an expensive double-and-add step instead. In practical applications, the Hamming weight of the seed is relatively small. In practical applications, the Hamming weight of the parametrized seed is relatively small, making the double-and-add algorithm more efficient.

Similarly, by acting $p^7$ on two sides of the above equation, the following formula

$$\left( \frac{Z_{[-z]Q}^{zp^6+1} \cdot v_Q(P)^{2p^8}}{Z_{[-z]Q+P}^{zp^6} \cdot Z_{[-z]Q+\sigma^{-1}(P)}} \right)^{\frac{p^{13}-1}{r}}$$

also gives a bilinear pairing. The idea in Algorithm 3 can also be leveraged to obtain the value $\left(\frac{Z_{[-z]Q}}{Z_{[-z]Q+P}}\right)^{zp^6} \cdot Z_{[-z]Q+\sigma^{-1}(P)}$. The computational process is stated in Algorithm 4. Denote by `xADD` the function of compatible addition.

According to Algorithm 4, we can see that it requires to perform one point doubling and two differential additions in a doubling iteration step. All these operations are performed in $\mathbb{F}_{p^{13}}$. It follows from Algorithms 7 and 8 that the costs for `xDBL` and `xDIFF` on $K = E/\langle \pm 1 \rangle$ over $\mathbb{F}_{p^{13}}$ are

$$\text{Costdbl} = 2\mathbf{m}_{13} + 3\mathbf{s}_{13}, \;\; \text{Costdiff} = 4\mathbf{m}_{13} + 2\mathbf{s}_{13}.$$

More precisely, some of the differential additions (Lines 5 and 15 in Algorithm 4) 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 8 the corresponding cost for this type of differential addition is $\text{Costdiff}_{\text{bit0}} = 3\mathbf{m}_{13} + 2\mathbf{s}_{13} + 13\mathbf{m}$.

As for the double-and-add step, we need to execute one compatible addition, together with three differential additions over $\mathbb{F}_{p^{13}}$. Since the coefficient $a$ is small, from Algorithm 9 the cost of `xADD` is about

$$\text{Costadd} = 11\mathbf{m}_{13} + 2\mathbf{s}_{13}.$$

On this basis, the computational cost for a basic iteration is

$$\begin{aligned} \text{Costcubic}_{\text{bit0}} &= \text{Costdbl} + 2\text{Costdiff}_{\text{bit0}} = 8\mathbf{m}_{13} + 7\mathbf{s}_{13} + 26\mathbf{m}, \\ \text{Costcubic}_{\text{bit1}} &= \text{Costadd} + 3\text{Costdiff} = 23\mathbf{m}_{13} + 8\mathbf{s}_{13}. \end{aligned}$$

## 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 utilizing Miller's algorithm and biextension. The cost calculations encompass the Miller iterations on families BLS12, AFG16, BW14 and BW13. Table 2 illustrates the 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.

**Algorithm 4** The optimized shared double-and-add ladder for computing $\left(\frac{Z_{[-z]Q}}{Z_{[-z]Q+P}}\right)^{zp^6} \cdot Z_{[-z]Q+\sigma^{-1}(P)}$ on BW13 family

**Input:** The points $-Q = (X_Q : Z_Q)$, $P = (X_P : Z_P)$, $-Q + P = (X_{-Q+P} : Z_{-Q+P})$, $-Q + \sigma^{-1}(P) = (X_{-Q+\sigma^{-1}(P)} : Z_{-Q+\sigma^{-1}(P)})$, $-Q - P = (X_{-Q-P} : Z_{-Q-P})$ 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 $z > 2$ that needs to be performed. Assume that $z = \sum_{i=0}^{N} n_i 2^i$, where $n_i \in \{0, 1\}$.

**Output:** The value $\left(\frac{Z_{[-z]Q}}{Z_{[-z]Q+P}}\right)^{zp^6} \cdot Z_{[-z]Q+\sigma^{-1}(P)}$

```
$R \leftarrow -Q,\ S_1 \leftarrow -Q + P$
$tab_1 \leftarrow [\ ],\ j \leftarrow 0$
for $i = N - 1$ to $0$ do
    if $n_i = 0$ then
        $R \leftarrow \texttt{xDBL}(R),\ S_1 \leftarrow \texttt{xDIFF}(S_1, R, iX_P)$
        $tab_1[j] \leftarrow R,\ j \leftarrow j + 1$
    else
        $T \leftarrow \texttt{xADD}(R, -Q, S_1, -Q - P),\ R \leftarrow \texttt{xDIFF}(T, R, iX_{-Q})$
        $S_1 \leftarrow \texttt{xDIFF}(T, S_1, iX_{-Q-P}),\ tab_1[j] \leftarrow T,\ j \leftarrow j + 1$
    end if
end for                                        ▷ $R = [-z]Q,\ S_1 = [-z]Q + P$
$j \leftarrow 0,\ c \leftarrow (Z_R/Z_{S_1})^{p^6},\ S_2 \leftarrow \left(c \cdot X_{-Q+\sigma^{-1}(P)} : c\right)$
for $i = N - 1$ to $0$ do
    if $n_i = 0$ then
        $S_2 \leftarrow \texttt{xDIFF}(S_2, tab_1[j], -iX_P),\ j \leftarrow j + 1$
    else
        $S_2 \leftarrow \texttt{xDIFF}(tab_1[j], S_2, c \cdot iX_{-Q-\sigma^{-1}(P)}),\ j \leftarrow j + 1$
    end if
end for                                        ▷ $S_2 = [-z]Q + \sigma^{-1}(P)$
return $c \cdot Z_{S_2}$
```

**Table 2** The computational costs of each iteration step of the Miller loop using biextension on families BLS12, AFG16, BW14 and BW13. There are two situations in each iteration step: bit $= 0$ and bit $= 1$.

| Family | bit $= 0$ | bit $= 1$ |
|---|---|---|
| BLS12, $D = 3$ | $\mathbf{m}_{12} + 2\mathbf{s}_{12} + 28\mathbf{m}_2 + 2\mathbf{s}_2$ | $6\mathbf{m}_{12} + 5\mathbf{s}_{12} + 58\mathbf{m}_2 + 5\mathbf{s}_2$ |
| AFG16, $D = 1$ | $2\mathbf{s}_{16} + 14\mathbf{m}_4 + 3\mathbf{s}_4$ | $5\mathbf{m}_{16} + 3\mathbf{s}_{16} + 35\mathbf{m}_4 + 3\mathbf{s}_4$ |
| BW14, $D = 1$ | $4\mathbf{s}_{14} + 14\mathbf{m}_7 + 3\mathbf{s}_7$ | $6\mathbf{m}_{14} + 5\mathbf{s}_{14} + 27\mathbf{m}_7 + 3\mathbf{s}_7$ |
| BW13, $D = 1$ | $8\mathbf{m}_{13} + 7\mathbf{s}_{13} + 26\mathbf{m}$ | $23\mathbf{m}_{13} + 8\mathbf{s}_{13}$ |

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 $\mathbb{F}_{p^k}$ ([?, Table 9] and [?, 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 Miller loop. 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 cost of exploiting the Miller's algorithm, we refer to [?, Table 7] and [?, 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 on families BLS12, AFG16, BW14 and BW13. Among them, the scenarios in which the biextension is proved to be more efficient are marked in red.

| Family | Approach | doubling | doule-and-add |
|---|---|---|---|
| BLS12, $D = 3$ | biextension | $214\mathbf{m}$ | $688\mathbf{m}$ |
| | Miller | $99\mathbf{m}$ | $170\mathbf{m}$ |
| AFG16, $D = 1$ [?] | biextension | $252\mathbf{m}$ | $900\mathbf{m}$ |
| | Miller | $200\mathbf{m}$ | $382\mathbf{m}$ |
| BW14, $D = 1$ [?] | biextension | $600\mathbf{m}$ | $1392\mathbf{m}$ |
| | Miller | $480\mathbf{m}$ | $954\mathbf{m}$ |
| BW13, $D = 1$ [?] | biextension | $1016\mathbf{m}$ | $2046\mathbf{m}$ |
| | 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 odd and the CM discriminant is 1, the computation of pairings by leveraging biextension will be more efficient. Consequently, the utilization of biextension for pairing computation holds practical application value in certain cryptographic scenarios.

# 5 Conclusion

In this work, we gave a detailed research for applying biextension to the pairing-based cryptography. The technique of biextension can be exploited to compute Tate pairing together with its variants. Overall, the efficiency of computing pairings using biextension is comparable to that of the Miller algorithm. In some specific cases, utilizing biextension 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 biextension algorithm, it will emerge as a competitive alternative to the Miller algorithm. The theory of biextension is also expected to have other applications in pairing-based cryptography.

# Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 12441107) and Guangdong Major Project of Basic and Applied Basic Research (No. 2019B030302008).

# References

# Appendix A The related algorithms

In this appendix, we present some associated algorithms required in the pairing computation through biextension, including $x$-only 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 5** $x$-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 $P$.

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

**Algorithm 6** $x$-only cubical differential addition on the curve $E : y^2 = x^3 + b$

```
Input: Two points $P = (X_P : Z_P), Q = (X_Q : Z_Q) \in E(\mathbb{F}_{p^k})$. The inverse of the
    $X$-coordinate of the differential of $P$ and $Q$: $iX_{P-Q}$.
Output: The coordinate $(X_{P+Q} : Z_{P+Q})$
$t_1 \leftarrow X_P + Z_P$
$t_2 \leftarrow X_P - Z_P$
$t_3 \leftarrow X_Q + Z_Q$
$t_4 \leftarrow X_P \cdot X_Q$
$t_5 \leftarrow Z_P \cdot Z_Q$
$t_6 \leftarrow t_1 \cdot t_3 - t_4 - t_5$
$t_7 \leftarrow t_2 \cdot t_3 - t_4 + t_5$
$X_{P+Q} \leftarrow (-4b \cdot t_5 \cdot t_6 + t_4^2) \cdot iX_{P-Q}$
$Z_{P+Q} \leftarrow t_7^2$
return $X_{P+Q},\ Z_{P+Q}$                    ▷ Total cost: $6\mathbf{m}_k + 2\mathbf{s}_k + 1\mathbf{m}_0$
```

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

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

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